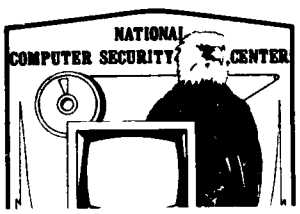
CSC-EPL-89/005

(2)



NATIONAL COMPUTER SECURITY CENTER

AD-A234 059

FINAL EVALUATION REPORT OF INTERNATIONAL BUSINESS MACHINES CORPORATION

VM/SP with RACF

28 September 1989

FINAL EVALUATION REPORT

INTERNATIONAL BUSINESS MACHINES CORPORATION

VM/SP WITH RACF

NATIONAL
COMPUTER SECURITY CENTER

9800 Savage Road
Fort George G. Meade
Maryland 20755-6000

28 September 1989

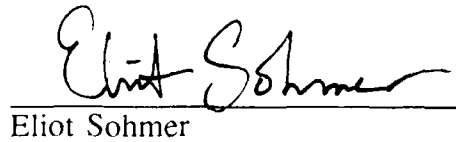
CSC-EPL-89/005
Library No. ~~S33,120~~
S233,120

This page intentionally left blank.

FOREWORD

This publication, the Final Evaluation Report International Business Machines Corporation, VM/SP with RACF, is being issued by the National Computer Security Center under the authority of and in accordance with DoD Directive 5215.1, "Computer Security Evaluation Center". The purpose of this report is to document the results of the formal evaluation of IBM's VM/SP with RACF operating system. The requirements stated in this report are taken from *DEPARTMENT OF DEFENSE TRUSTED COMPUTER SYSTEM EVALUATION CRITERIA* dated December 1985.

Approved:



28 September 1989

Eliot Sohmer
Chief, Office of Product Evaluations
and Technical Guidelines
National Computer Security Center

ACKNOWLEDGEMENTS

Team Members

Team members included the following individuals, who were provided by the following organizations:

R. Leonard Brown, Ph.D.

The Aerospace Corporation
El Segundo, California

Kenneth Dixon Vane
David L. Gill

MITRE Corporation
McLean, Virginia

Michael J. Oehler
Robin A. Willinghan

National Computer Security Center
Ft. George G. Meade, MD

Further Acknowledgements

Initial contributions were given by James L. Arnold, Jr., Bruce Crabill, Mark Gabriele and Rick A. Siebenaler.

CONTENTS

EXECUTIVE SUMMARY

The security protection provided by the International Business Machines Corporation VM/SP with RACF operating system software as described in Appendix B, running on one of the IBM 370 processors listed in Appendix A and configured in an appropriately trusted manner as described in the Trusted Facilities Library[28] has been examined by the National Computer Security Center (NCSC). The security features of VM/SP with RACF were examined against the requirements specified by the DoD Trusted Computer System Evaluation Criteria [40](the Criteria) dated 26 December, 1985 in order to establish a candidate rating.

The NCSC evaluation team has determined that the highest class at which VM/SP with RACF satisfies all the specified requirements of the Criteria is class C2.

A system that has been rated as being a C division system provides for discretionary (need-to-know) protection and, through the inclusion of audit capabilities, for accountability of subjects and the actions they initiate. Such a system is expected to run in an environment of cooperating users processing data at the same level of sensitivity.

Systems in this class enforce discretionary access control (DAC) at the granularity of single users, making each user accountable for his actions through login procedures, auditing of security relevant events, and resource isolation.

The VM/SP with RACF operating system has a number of features that enhance the security of the computing system. A summary of these features can be found at the beginning of the System Overview section (see page 5, "System Overview").

This page intentionally left blank.

INTRODUCTION

In March of 1987, the National Computer Security Center (NCSC) began a product evaluation of a computer system consisting of IBM 370 architecture hardware running the VM/SP operating system software, supported by the Resource Access Control Facility (RACF) and other software products. This system, which will be called VM/SP with RACF in this report, provides for separation of individual users and their data to the extent that each user appears to have exclusive access to an entire IBM 370 machine. In addition, the system allows for controlled sharing of executable code used in common by multiple users, and of data which individual users wish to share at their discretion. It is intended that this report give evidence and analysis of the security features and assurances provided by VM/SP with RACF. This report documents the evaluation team's understanding of the product's security design and appraises its functionality and integrity against the C2 level requirements in the *Trusted Computer System Evaluation Criteria*[40].

This evaluation applies to VM/SP Release 5 or VM/SP HPO Release 5, each with the C2 Security Function modification applied, each of which comes with CMS Release 5, together with RACF Release 1.8.2, Virtual Machine/Directory Maintenance Licensed Program Release 1.4, ISPF Release 2.2 and VMTAPE-MS Release 4.1. Virtual Machine/Directory Maintenance Licensed Program became available on 31 March, 1989. The remaining products have been available since fourth quarter of 1988.

Material for this report was gathered by the NCSC evaluation team through documentation, training and interaction with system developers.

Evaluation Process Overview

Background

The National Computer Security Center (NCSC) was created to improve the state of computer security in computer systems processing information that is vital to the owners of that information. The Center fulfills its mission by promoting the development and implementation of Trust Technology and encouraging the widespread availability and use of trusted computer security products. Through the Trusted Product Evaluation Program, the Center works with the manufacturers of hardware and software products to implement and make available to the public technically sound computer security solutions. Under this program, the NCSC evaluates the technical protection capabilities of computer security products against well-defined published evaluation criteria.

The product evaluation process culminates in the publication of a Final Evaluation Report, of which this document is an example. The Final Evaluation Report describes the product and assigns it a rating that denotes a specific level of trust. The asssigned rating is independant of any consideration of overall system performance, potential applications, or particular processing environment. Rated products are listed on the Evaluated Products List (EPL), the aim of which is to provide ADP system developers, managers, and users an authoritative evaluation of a product's suitability for use in processing important information.

The NCSC performs evaluations of computer products in varying stages of development from initial design to those that are commercially available. Product evaluations consist of a developmental phase and a formal phase. All evaluations begin with the developmental phase. The primary thrust of the developmental phase is an in-depth examination of a manufacturer's design for either a new trusted product or for security enhancements to an existing product. Since the developmental phase is based on design documentation and information supplied by the industry source, it involves no "hands on" use of the system. The developmental phase results in the production of an Initial Product Assessment Report (IPAR). The IPAR documents the evaluation team's understanding of the system based on the information presented by the vendor. Because the IPAR contains proprietary information, distribution is restricted to the vendor and the NCSC.

Products entering the formal phase must be complete security systems. In addition, the release being evaluated must not undergo any additional development. The formal phase is an analysis of the hardware and software components of a system, all system documentation, and a mapping of the security features and assurances to the Criteria. The analysis performed during the formal phase requires "hands on" testing (i.e., functional testing and, if applicable, penetration testing). The formal phase results in the production of a final report and an Evaluated Products List entry. The final report is a summary of the evaluation and includes the EPL rating which indicates the final class at which the product successfully met all Criteria requirements in terms of both features and assurances. The final report and EPL entry are made public.

Document Organization

This report consists of four major sections and five appendices. Section 1 is an introduction. Section 2 provides an overview of the system hardware and software architecture. Section 3 provides a mapping between the requirements specified in the TCSEC and the system features that fulfill those requirements. Section 4 presents pertinent facts about the system features that are in some way notable but do not otherwise have a place in this document. The appendices identify the specific hardware and software components to which the evaluation applies, the specific systems which the team tested, including the added tests the team performed to supplement the vendor tests, and also contain a table of references and a Glossary of terms.

Conventions

In the text, the symbol [n] refers to a document in Appendix C. If the symbol appears following an entire paragraph, then all of the material from that paragraph has been taken from the document referred to.

The names of software modules, down to the level of individual assembly language components, will be referred to in all capital letters. Thus, CP refers to the entire set of VM/SP Control Program software, while DMKIAC refers to a single module within CP.

The name of a Virtual Machine, whether it is one initialized for a user or is a service virtual machine (SVM) that is part of the Trusted Computing Base, will be followed by /VM. Thus,

the RACF/VM is the RACF service machine, while JONES/VM refers to a virtual machine for user JONES. In some instances, the actual name of a SVM ends in VM. In such a case, the superfluous /VM will not be included. By itself, VM stands for virtual machine.

Other acronyms are listed in the Glossary, which is appendix D of this report.

This page intentionally left blank.

SYSTEM OVERVIEW

The architecture of the IBM 360 and its successor, the IBM 370, has proved to be so suitable for such a wide variety of applications that the largest number of mainframe systems in the world meet the specifications for this architecture. Unlike the majority of systems, which have exactly one operating system for each architecture family, and in contrast to the Unix[1] paradigm in which the same operating system runs on numerous different micro, mini and mainframe computers, several different operating systems have been developed to run on IBM 370 computers.

Long before desktop microcomputing was developed International Business Machines Corporation decided there was a need to allow individual users to have the effect of having the use of an entire IBM 370 without actually tying up a complete set of hardware. Initially this allowed developers of several operating systems that would all run on the same equipment to perform both development and maintenance programming without the inconvenience of a system crash denying service to all other users. This concept of a virtual[2] machine has found wider application in education and business, and the Virtual Machine/System Product (VM/SP) which allows each user to have a desktop terminal act as if it is the only one connected to a mainframe computer is installed in a large number of sites.

At first inspection, it would seem that security would not be a problem on such a system. When a user logs on, the Control Program (CP) sets up a virtual machine that connects the user's real terminal to a virtual CPU, running in virtual memory, with virtual card punch, card reader and printer attached to each virtual machine (VM).

The user may choose to run any operating system that executes on the IBM 370 architecture, or to run a stand alone application program. For other than operating system development, most users choose to run the Conversational Monitor System (CMS). However, other uses for the system are to run applications that were developed for older operating systems such as VSE, or to provide the ability to run a large scale timesharing and batch operating system such as MVS[3] and maintain the ability to load and test configuration changes to the system in a separate VM without bringing the entire system down.

---

[1] Unix is a trademark of AT&T Bell Laboratories

[2] DEFINITION: virtual - adj. being so in effect or essence, although not in actual fact or name.

[3] When running a multi-user MVS, users must use the DIAL command to initiate login with the MVS virtual machine. The DIAL command, and thus multi-user operating systems running within a single VM, are excluded from the evaluated configuration. See the section "Disabling the DIAL and MESSAGE Commands" in the Security Guide [28].

In addition, the user's VM is connected to a number of minidisks, each of which is a fixed partition of the system's real disk storage. The security problem arises when one user asks to connect to a minidisk that belongs to another user. Such sharing of disk memory, originally controlled by sharing a password between one user and the other users trusted to access the minidisks, causes the same security problem encountered in a multiuser system with its disks divided into files. The owner or creator of each file wants to be able to decide which other users may have access to these files. Passwords are not a good way to provide this discretionary access control. In particular, it is never possible to list, for any file or any user, what the access capabilities are.

To provide more finely grained discretionary access control than that provided by passwords, IBM decided to adapt the Resource Access Control Facility (RACF), which also executes on its MVS operating system, to VM/SP. RACF allows all users and objects to be defined to the system, and then VM/SP consults with the RACF service machine, a VM that is always running whenever a user is logged on, to determine whether to allow a requested access.

One of the system tables that CP depends on is the CP Directory which describes for each user the default virtual machine that may be set up for that user. Since the Directory is so important for the proper functioning of the system, and because the format of the individual records is so important, the Virtual Machine/Directory Maintenance Licensed Program, commonly called DIRMAINT after its service virtual machine's name, was developed to allow the system administrator to safely initialize and modify entries in this Directory.

To ensure that information in the CP Directory is consistent with information used by RACF, the Interactive System Productivity Facility (ISPF) is used to provide menu driven functions for adding and changing security relevant records in the CP Directory that are used by both CP and RACF.

Since it is also possible to have tape drives attached to the real system, VM/SP must be supplemented with a system to keep track of access rights to individual tapes. The software for this system, called VMTAPE-MS, assures users that information is properly transferred from VM to tape and tape to VM.

It is possible for an administrator to log onto service virtual machines, such as those that run with RACF or VMTAPE. When one does this, CMS is used to interact with the virtual machine, so the copy of CMS used by the SVMs must also be trusted to run correctly.

As can be seen from the above descriptions of the components of this trusted system, there exist a number of possible interfaces between users and the software portions of the Trusted Computing Base. When a user logs on, a virtual machine is created that can run any System 370 software ever written. At this point the user can perform an IPL (initial program load) of such software or the user could have arranged for such an IPL to occur by default at logon time. Once such an IPL has occurred, the user interacts directly with that software.

However, the user is also able to interact with CP either directly by typing in commands, or indirectly by running programs that interact with CP through DIAGNOSE codes. In the former instance, if the user is either running CMS or if the user puts the terminal into CP

READ state by using an appropriate Program Function (PF) key, any command which the user has authorization to execute can be entered from the terminal. These commands are listed in *CP Command Reference* [22].

In the latter instance, there is a privileged IBM 370 instruction X'83' DIAGNOSE which, if issued by a program running on real hardware, performs a hardware self-test. When issued by a virtual machine, the DIAGNOSE instruction causes a program interrupt which returns control to CP. The DIAGNOSE instruction has room for a 16 bit code, and IBM has assigned codes X'00' through X'FC' to specific functions that CP can perform for a virtual machine. *VM/SP System Facilities for Programming* [27] contains the description of all these DIAGNOSE codes. In particular, DIAGNOSE code X'08' includes as an argument a pointer to a string of characters. This string corresponds to a CP command to be executed. Thus, any program running in any virtual machine can issue any command that the user could issue by using this DIAGNOSE code.

In the following sections, the history and architectural detail of this trusted system will be described. The first section is a history of RACF, VM/SP, CMS, DMPP, ISPF and VMTAPE-MS. The following section will describe the system 370 hardware architecture. The next section will describe the individual software products that make up the Trusted Computing Base (TCB) software, together with the trusted VM's that assist in securing the system. Following that, the system subjects and objects are identified, and then the protection mechanisms that are used to implement the security policy of the VM/SP with RACF trusted system are reviewed, making reference to the tasks of the individual TCB components.

History of VM/SP Trusted System

The development of the first operational Compatible Time Sharing System (CTSS) took place in the early 1960's. CTSS was in production and used at MIT from 1964-1974. CTSS had the greatest influence upon VM/SP, which was initially known as CP.

Research for CP began about 1965 and in 1966, it became operational on a S/360 Model 40. This system, CP/40, used the Cambridge Monitor System (CMS) as its operating system. CP/40 was renamed CP/67, which was built for the S/360 Model 67. By the late 1960's, CP/67 was running at Cambridge, MA and MIT's Lincoln Lab. In the early 1970's, CP/67 was modified to provide a virtual 370 machine and second level virtual 370 machines on a S/360-67 model CPU.

VM/370 was announced in 1972 by IBM. This was a simulation of multiple System 370 systems (virtual machines) within a single hardware system. There have been 6 releases of VM/370 announced throughout the 1970's, each adding various hardware support and functional enhancements. However, it was not until 1980 that VM became recognized as a commercial product.

VM/SP Announced

VM/SP (Virtual Machine/System Product) was announced in 1980. There have been 6 releases of VM/SP, each consisting of additions to improve the overall system performance.

VM/SP 1 introduced a new editor (XEDIT), InterUser Communication Vehicle (IUCV) (see page 41, "IUCV"), EXEC2 macro language, and PROFS. In March 1982, VM/SP 2 added the extensive table lookup message controls (RTABLES), other features of Programmable Operator (PROP), and the Productivity Aids group of tools. In 1983, VM/SP 3 introduced REXX which superceded EXEC2, XEDIT enhancements, and SQL/DS was supported providing relational database ability. VM/SP 4 was announced in 1984. It improved ACCESS performance, file searching performance, and support for DBCS characters was added to XEDIT. VM/SP 5 was announced in October 1986. It added the mixed case error messages, "full screen" CMS, Transparent Services Access Facility (TSAF) which provided cross-CPU abilities and Advanced Program to Program Communication (APPC/VM) (see page 43, "Advanced Program to Program Communications") which embodied the standard System Network Architecture (SNA) protocol for program to program communications. In October 1987, VM/SP 6 was announced. It introduced Shared File System (SFS) and APPC VTAM Support (AVS). Approximately 2 years after VM/SP, VM/SP HPO (High Performance Option) was announced.

VM/SP HPO provides extended features to the VM/SP control program. VM/SP HPO adds hardware support improvements, microcode assists, and performance, operational and Reliability/Availability/Serviceability (RAS) enhancements. It primarily provided significant performance enhancements for CMS intensive environments on 3081s. VM/SP HPO 2 added enhancements for running MVS in second level. Support for new high-speed paging devices was added with VM/SP HPO 3. HPO 3.6 added 3090 support with HPO 3.4 features. In December 1985, HPO 4.0 was announced. It was the HPO 3.4 function on top of VM/SP Rel.4. HPO 4.2 was the combination of HPO 4.0 with SP4 code and HPO 3.6 code. Support for the VECTOR function was also added. VM/SP 5 HPO was announced in October 1987 and has been available since Fall of 1988. It provided relief from the 9900 system spool file limit and given sufficient DASD, each user may have 9900 spool files. Although VM/SP HPO provides additional enhancements, it offers similar functional capabilities to VM/SP.

Other Products in VM/SP with RACF

Additionally IBM has developed the trusted system programs which, when used with the operating system, serve and assist both the system and its users. Each one of these programs has a dedicated task. A brief background is given here for these products: Conversational Monitor System (CMS), Resource Access Control Facility (RACF), Virtual Machine/Directory Maintenance Licensed Program (DIRMAINT), Interactive Systems Productivity Facility (ISPF), and VMTAPE.

CMS

The Cambridge Monitor System (CMS) was introduced in the early 1970's. It was the predecessor of the CMS (Conversational Monitor System) we know today which was announced right after the Cambridge Monitor System. The prior code called VM is now called Control Program (CP). CP directs the hardware, and CMS is the operating system that each user runs. Each user thinks that he has the system, but instead CP is simulating the entire hardware environment for each CMS system running.

RACF

Resource Access Control Facility (RACF) is a security enhancement. RACF provides identification and authentication, access control, and accountability mechanisms. It was initially developed as an MVS product. On June 25, 1984, limited availability of the first release of RACF/VM was announced. This used RACF/MVS version 1 Release 6 code augmented with the support code which interfaces the RACF MVS product to the VM environment. RACF 1.7 plus the VM specific support code became a Program Product on October 7, 1986. Both RACF 1.8.2 and RACF 1.8.3 were made available in the Fall of 1988.

DIRMAINT

Directory Maintenance Program Product (DIRMAINT) R1 was announced in December 1979. It was the original DIRMAINT product which provided a command driven mechanism for updating both the source directory and the online form of the directory. The general user was allowed to make certain changes which did not impact system resources, and the privileged user could use certain commands to update the directory rather than having to EDIT the source directory file. It also allowed the DATAMOVR virtual machine to remove data from a minidisk after it was deleted or copy the data from a minidisk to a new version. This version of DIRMAINT could be used with various levels of VM/370.

DIRMAINT R2 was announced in October 1982. It included updating of the DATAMOVR cleanup and copy functions to allow them to be automatically invoked. R2 also added the DIRM SCREEN command which modifies SCREEN directory control statements. The DIRM IPL command was extended to allow specification of the PARM option, and the DIRM SETOPTN command was added to control the interaction between a user and the DIRMAINT virtual machine.

In March 1988, R3 was announced. VM/XA SP directory was supported which included new functions available to process VM/XA unique directory control statements. The ability to issue DIRMAINT commands in a virtual machine running CMS in a XA mode was also supported.

R4 will be announced with this evaluation. It significantly improves performance of DIRMAINT in terms of how much time and resource is necessary to make a change to the source directory. It removes some functional restrictions which put limits on the size of the directory and the size of an individual user definition it could handle. It automatically changes or deletes LINK statements, adds a new DIRMAINT command privilege class which allows the virtual machine to execute commands which can look at the directory and status of DIRMAINT but not change them. The DIRMAINT help facility now uses the standard CMS HELP facility and adds new user exits. R4 adds a minidisk password auditing facility along with the capability of having DIRMAINT generate a new random logon

password for all users when requested. It adds a BATCH command submission facility[1], DIRM ADD LIKE prototype facility, query functions to several general user commands, and several new general user and privileged user commands for usability purposes.

The current version of this software is referred to by the vendor as Virtual Machine/Directory Maintenance Licensed Program but in this report the term DIRMAINT will be used for the software when it is clear what is being referred to.

ISPF

The Structured Programming Facility (SPF) began as a program development tool in the TSO environment and was designed to take advantage of the characteristics of 3270 Display Terminals. Facilities were provided to increase productivity for users of both structured and conventional programming techniques. This level of SPF (sometimes called "old SPF") was also provided for the VM/CMS environment and was functionally equivalent to the SPF/TSO product. In 1980 "old SPF" was extended to include Dialog Manager support. The Dialog Manager provided new functions that simplified the development of interactive applications. At this time, the product was renamed System Productivity Facility ("new SPF"). In 1982 the SPF product was further enhanced and also split into two products; Interactive Systems Productivity Facility (ISPF) and the ISPF Program Development Facility (ISPF/PDF). ISPF consisted of all the Dialog Manager support and ISPF/PDF ran as an application under ISPF and consisted of those functions that assist program development. ISPF/DM is the Dialog Manager part of ISPF. A dialog is any interactive session between a user and the system. ISPF/DM provides the following services: select, display, message, table, file, and variable. ISPF/DM also provides the user with: a consistent interface, online tutorial/help facilities, error recovery, a set of system commands, and the ability to have application and function commands.

VMTAPE-MS

VMTAPE-MS (Management System) permits VM installations to manage the usage of tape drives and tape volumes. VMTAPE-MS 1.2.1 became available on February 22, 1985. Its key features are its extensive tape librarian facilities, traceability of tape access, determinability of the date when a volume can be scratched, and reduced operator requirements for tape drive and volume control.

During 1986, VMTAPE-MS 1.3.1 became available. Product support for the IBM 3480 tape drive was a key new feature, along with a new utility, VMTDSP, which allows users to update the message display on IBM 3480 tape drives. Shared Tape Allocation Manager (STAM), a new facility, provides for the logical sharing of tape drives between multiple VM/SP operating systems. The automatic scratch volume selection facility (AUTOPICK) enables VMTAPE to select a specific scratch volume to satisfy a scratch mount request. The

---

[1] No Batch facilities are included in the evaluated product, so this facility can not be taken advantage of.

SETUP facility allows volume staging messages to be forwarded to a remote tape operator. Four digit real device addresses are fully supported and tape drives can now be defined on channels above channel 15.

The availability date for VMTAPE-MS 1.4.1 was November 25, 1988. The significant features of this product are that it can reserve one or more tape drives for a batch job, it supports simulation enhancements for multivolume processing and it adds interface enhancements.

IBM has been developing VM for the past twenty to twenty five years. Many new capabilities, applications, and performance factors have been incorporated.

Hardware Architecture

The hardware base for VM/SP with RACF is the IBM System/370. This evaluation does not include systems operating in the 370-XA mode.

Processors with a 370-XA mode must be set to System/370 mode (non-XA) using the architectural-mode-selection controls before VM/SP with RACF can be loaded and run. VM/SP with RACF will not load in 370-XA mode.

This section describes the hardware features of VM/SP with RACF.

Summary of System Architecture

The System/370 consists of main storage, one or more Central Processing Units (CPUs), channel sets, channels, and Input/Output (I/O) devices. I/O devices are typically attached to channels through control units. The logical structure of a single CPU System/370 is given in Figure 2.2.1. A system viewed without regard to its I/O devices is called a configuration.

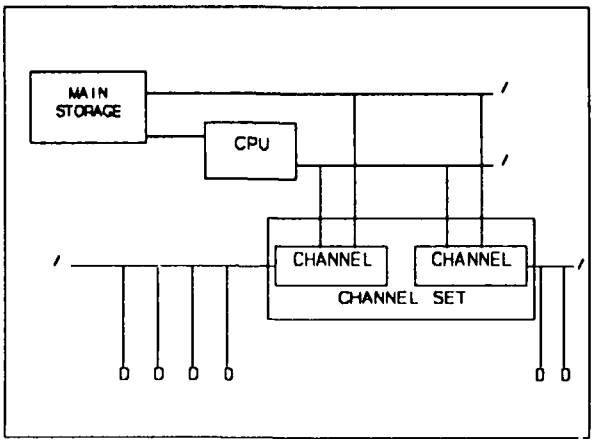


FIGURE 2.2.1

The System/370 incorporates the following facilities: Dynamic Address Translation (DAT), hardware protection, extended real addressing, multiprocessing, extended precision floating point, channel set switching, timing facilities, and program event recording. Extended real addressing is supported with the High Performance Option of VM/SP with RACF. The other facilities will be described in the paragraphs that follow.

Operator facilities are also provided for initial program load (IPL), maintenance operations, and diagnostic operations.

Appendix A contains a list of the processors that can be contained in a VM/SP with RACF configuration. It also identifies controllers and devices that can be connected to these configurations. Two separate lists are given in Appendix A. The first list indicates processors with controllers and devices that support VM/SP with RACF without the High Performance Option (HPO). The second list indicates processors with controllers and devices that support VM/SP with RACF with HPO. The High Performance Option is described on page 49.

The 3084 processor and several of the 3090 processors support partitioned processing, in which two separated 370 configurations exist within a single main-frame. No information connections may exist between the two configurations. The systems must be configured so that all devices are non-shared between the systems. If either system is to run the evaluated system, it must configure and load separate copies of VM/SP with RACF for each partition. For most of the processors that can be partitioned, another form of partitioning, called logical partitioning, is also possible. However, logical partitioning is disallowed for processors given in Appendix A.

Although the hardware supports various types of network interconnection devices, and software exists to take advantage of this hardware, none of these devices are allowed in the evaluated configuration.

Main Storage

Main storage provides high-speed processing of data by the CPUs and channels. The unit of memory addressable data is an eight bit byte. Two byte, four byte and eight byte contiguous memory areas can be referenced as data units. These data units are respectively called halfword, word and doubleword data units. Contiguous bytes of storage can be referenced as character data.

For purposes of address translation, and to simplify discussions about prefixing, storage protection and large units of data, another unit of data, called the page, will also be defined here. A page is a block of fixed size storage at sequential addresses . The address of the page is the lowest address of the block. The address of the page must be a multiple of the page size. For VM/SP with RACF the page size is 4,096 (4K) bytes.

Hardware support is also provided for segments. A segment is a not necessarily contiguous block of up to 16 pages of virtual memory. The base address of each page in the segment is stored contiguously in a segment table along with other information about the status of the page. The segment table is maintained by the system software and used by the system

hardware during dynamic address translation of virtual memory pages. See the discussion of system tables on page 34 for further information.

The base for addressing and arithmetic is base two (2). Address locations whose absolute storage address are multiples of two (2**1), four (2**2), or eight (2**3) are identified as base 2 integral boundary locations for halfword, word, and doubleword data, respectively. Although most instructions will fetch multiple byte data located at any storage address, significant efficiencies in retrieval time or storage time occur when multiple byte data units are allocated at an integral boundary location for the particular data type. An extended floating point option provides for double-doubleword data with integral boundaries occurring at absolute storage addresses which are multiples of sixteen (2**4).

For most operations, accesses to storage proceeds from the lowest address to the highest address within an addressed unit of data. The address of data or an instruction refers to the lowest byte containing that data or instruction. Addresses are twenty four bit unsigned integers, which provide 16,777,216 (16M) byte addresses.

Main storage may include one or more smaller faster access buffer storage areas called caches. A cache is usually physically associated with a CPU or an I/O processor. Cache usage is transparent except for the effect on performance.

CPU instructions and I/O operations can reference main storage concurrently. If a concurrent request to a main storage location occurs, access is normally granted in a sequence that assigns highest priority to channels and then to active CPUs.

For purposes of addressing main storage, three basic types of addresses are recognized: absolute, real, and virtual. An absolute address is an address assigned to a main storage location. Absolute addresses are used for storage access without any further address transformations. Address translation is described in the following section.

A real address is used in a multiple CPU configuration. In a multiple CPU configuration, two or more CPU units share the same main storage. Each CPU processes as though the first page of main storage belonged exclusively to that CPU. This area of main storage contains interrupt vectors and other control information. This initial page is referred to as the prefix area. Each CPU has a prefix register which is used to map real addresses to absolute addresses. The prefix register identifies the page in main storage which is swapped with the first page during translation of real addresses to absolute addresses. The actual bytes in the affected pages are not interchanged, only the corresponding addresses. In a single CPU configuration, absolute addresses equal real addresses.

A virtual address identifies a location in virtual storage. For a process, virtual storage consists of a set of pages which resides on external storage and a more up-to-date copy of some of these pages, called the working set, which resides in main storage. The pages map to an address space which usually exceeds the available main storage. Only the most recently active pages reside in main storage. When a virtual address is used for an access to main storage, it is translated by means of dynamic address translation to a real address, which is then converted to an absolute address by prefixing. If the virtual address references a page which

is not in main storage, the page is retrieved from external storage and made available in main storage. When dynamic address translation is disabled, real addresses equal virtual addresses.

Processor Description

The CPU of a System/370 configuration is the control center of the system. A CPU processes five types of instructions: general, decimal, floating-point, control and input/output. In performing its functions, it uses internal storage that is not part of main storage.

A CPU provides special internal storage registers. Registers can be addressed by instructions but are separate from main storage. These registers include the program status word (PSW), sixteen (16) general purpose registers, four (4) floating-point registers, sixteen (16) control registers, a prefix register, a clock comparator and the CPU timer. One primary performance characteristic of registers is that they are stored within the CPU, saving a main storage cycle when referenced.

Program Status Word

The program status word (PSW) includes the instruction address, condition code, and status information used to control instruction sequencing and to determine the state of the CPU. The active or controlling PSW is called the "current PSW." It governs the program currently being executed. When interrupts occur the current PSW is stored so that execution can resume at the point of interrupt.

A PSW contains sixty-four (64) bits of information. This information is stored and interpreted based on the value of the EC mode bit, bit 12 of bits 0 through 63. The two control modes are the extended control (EC) mode (PSW bit 12=1) and the basic control (BC) mode (PSW bit 12=0.)

EC mode takes full advantage of the System/370 architecture. This includes dynamic address translation using control register 1. The Control Program (CP) software component of VM/SP with RACF executes in EC mode but with dynamic address translation disabled. All other software components of VM/SP with RACF execute in EC mode with dynamic address translation enabled. The CP component manages segment and page tables for the rest of the processes executing in EC mode.

BC mode corresponds to the PSW structure in the IBM System/360 architecture. This architecture has no dynamic address translation. All references to main storage for a process executing in BC mode are absolute. A guest operating system running in a VM can use BC mode.

Bit 15 of both modes is used to determine in which of two states the CPU is executing. When bit 15 is zero, the CPU is in the supervisor state. In the supervisor state, all instructions are valid. When bit 15 is one, the CPU is in the problem state. In the problem state, only those instructions are valid that provide meaningful information to an application program and that cannot affect the system integrity. Instructions that can only execute in supervisor state are called privileged. All other instructions are called unprivileged

instructions. When a CPU in the problem state attempts to execute a privileged instruction, a privileged-operation exception is recognized. A CPU in privileged state can execute any instruction, privileged or unprivileged.

Bits 8-11 form the PSW key. This key is the access key for main storage references by the CPU. If the PSW access key is zero, write and fetch access to all main storage locations is granted. Each half page of main memory (2,048 byte blocks) has a storage key associated with it. The storage key is seven bits in length and contains a four bit access-control field, a fetch-protection bit, a reference bit, and a changed bit. The referenced and changed bits will be discussed later under dynamic address translation. If the PSW access key is non-zero, then write access is granted to a byte in main storage only if the associated storage key contains an access-control field value matching the PSW access key. The fetch-protection bit controls fetch access when the PSW access key is non-zero. If the access key from the PSW is non-zero and the fetch- protection bit of the applicable storage key is zero (0), then fetch access is allowed to the byte in main storage regardless of the access-control field value in the storage key. If the access key from the PSW is non-zero and the fetch-protection bit is one (1), then fetch access is allowed to the byte in main storage only if the access-control field value of the associated storage key matches the PSW access key.

Other fields in the PSW contain the following:

- a program event recording mask
- results of executing certain instructions (condition code)
- arithmetic instruction exception masks
- an indicator that primary and secondary address spaces are available for use (This feature called dual-address space is not used or supported in VM/SP with RACF.)
- the address of the leftmost byte of the next instruction to be executed
- a mask for I/O interrupts
- a mask for machine-check interrupts
- a mask for external interrupts.

Fields that are not defined in the PSW are set to zero.

Interrupt Processing

The CPU has an interrupt capability which permits the CPU to switch its state and process another program in response to conditions external to the configuration, within the configuration, or within the CPU itself.

When an interrupt occurs, the CPU places the current PSW in an assigned storage location, called the old-PSW location. The CPU fetches a new PSW from a second assigned storage location. This new PSW is used to set the state of the CPU and to determine the next instruction to execute. The usual intent of the program referenced by the new-PSW is to respond to the interrupt which has just occurred.

When it has finished processing the interrupt, the interrupting program may reload the old PSW and return control to the interrupted program.

Another use of the exit from an interrupt handler is made with the CP software component of VM/SP with RACF. When interrupt processing is complete, transfer is made to a CP dispatcher module which provides fair scheduling of virtual machines ready to execute. See page 36, "Dispatch and Scheduling".

There are six classes of interrupts: external, I/O, machine check, program, restart, and supervisor call. Each class has a distinct pair of old-PSW and new-PSW locations permanently assigned in main storage. The current PSW may contain bits that mask any of the three interrupts: external, I/O, or machine check. If one of these interrupts has occurred and the interrupt is being handled with subsequent interrupts of this type masked, then subsequent interrupts will, in general, remain pending until completion of the interrupt. Each type of interruption has a two to eight byte interruption code which is stored in low main storage when the interrupt is handled. The location of the interruption code is fixed, but differs for each class of interrupt.

When external interrupts are not masked by the current PSW (bit 7=1), more than one source may present a request for external interrupt at the same time. When this happens, the external interrupt(s) processed will be the one(s) with the highest priority. External interrupts are listed below in highest to lowest priority:

- Interval timer decremented to or through zero, interrupt key activated by operator, signal received on one or more of the six signal-in lines (Multiple conditions are indicated concurrently, and handled as one interrupt.)
- Another CPU in the configuration has lost power or entered the check-stop state
- Emergency signal from SIGNAL PROCESSOR instruction executed by this CPU or another CPU in the configuration
- External call from SIGNAL PROCESSOR instruction executed by this CPU or another CPU in the configuration
- Time-of-day (TOD) clock synchronization check for multiple-CPU configuration
- Clock comparator less than compared portion of TOD clock
- CPU timer is negative

- Configuration control or maintenance functions have completed (These are model-dependent.)

During the execution of an instruction, several interruption-causing events may occur simultaneously. Simultaneous interruption requests are honored in a predetermined order.

Machine-check interrupts determined to be exigent have the highest priority. Exigent machine-check interrupts indicate that damage has or will occur such that execution of the current instruction or of the current interruption sequence cannot safely continue. If processing were to continue either instruction processing damage or system damage would result.

If multiple interruption requests exist concurrently at any point where interrupts can be processed and multiple classes of interrupts are present in the requests and no exigent machine-check interrupts are included, then the multiple interrupts will be processed in the following interrupt class order: supervisor call, program, non-exigent machine check, external, i/o, and restart.

Interruption requests are processed by storing the current PSW in the old PSW belonging to the highest priority interruption present and fetching the new PSW of that interruption. As long as concurrent interrupts remain, the new PSW is immediately stored without execution of any instructions in the old PSW of the next lower priority interrupt present and the new PSW of the next lower priority interrupt is loaded. This process continues until all interruptions are stacked for servicing. CP processing then continues at the program counter value in the last PSW fetched, which has been set to the location of the interrupt handler for the highest class interrupt currently active. This implies that execution of queued interrupts will occur in reverse order of the interrupt classes listed in the preceding paragraph.

General Registers

Sixteen general registers may be used as accumulators in general arithmetic and logical operations. Each register contains 32 bits. These are referenced in four bit fields within instructions. For some instructions, two adjacent general registers can be paired for processing as a 64 bit string. In these instructions, an even register value references the even register and the next odd register as an even-odd pair. All except the lowest general register, register 0, can also be used in address arithmetic (see page 22, "Instruction Set").

Floating Point Registers

Four floating-point registers provide storage and processing arithmetic for floating-point data. Each register is 64 bits in length. Floating-point instructions deal either with the complete 64 bit string or the left 32 bits. For the shorter floating-point string, only precision is lost in computations. In cases where even better precision is required, the first two or the last two floating-point registers can be paired for extended precision computations.

Control Registers

Sixteen control registers, each having 32 bit positions, are available through the LOAD CONTROL and STORE CONTROL instructions. Like the general registers, they are identified by the numbers 0-15, represented by four bit R fields of these instructions. The bit positions of the control registers are assigned to particular facilities in the system, such as program-event recording, and are used either to specify that an operation can take place or to furnish special information required by the facility. Multiple control registers can be addressed by the LOAD CONTROL and STORE CONTROL instructions.

Control registers 0 and 1 are used during dynamic address translation (DAT). Five bits in control register 0 determine the page size and segment size for DAT. Control register 1 contains the eighteen high-order bits of the segment table origin. Bits 8-25 of control register 1, with six zero bits appended on the right, form a twenty-four bit real address that designates the beginning of the segment table to use for DAT. Control register 1 also contains a value used during DAT to screen invalid segment-index fields in a virtual address. The name of this field in control register is the (primary) segment table length. It is in bits 0-7. The size of the segment table referenced by control register 1 can be determined from the formula:

segment table size = ((value of bits 0-7 of control register 1) + 1) * 64.

Since each entry in the segment table is four bytes long, each segment table will consist of multiples of sixteen segment table entries. Bits 0-7 of control register 1 will be zero for 1M segment sizes and all segment-index fields will reference one of the sixteen entries in the segment table. For 64K segment sizes, a segment-translation exception is recognized when bits 0-7 of control register 1 is less than the eight-bit value formed by appending four zeros to the left of bits 8-11 (the four high-order bits of the segment-index) of a virtual address.

Address Translation by Prefixing

This section and the next describe the two types of address translation that occur in transforming a virtual address to an absolute address: prefixing and dynamic address translation.

Prefixing is used whenever multiple CPUs exist within a System/370 configuration. In a single CPU configuration, the absolute addresses 0 through 4095 contain critical control information such as new and old PSWs for each of the classes of interrupts.

In a multiple-CPU configuration, each CPU views its real addresses 0 through 4095 as the area where interrupts are handled. Each CPU in a multiple-CPU configuration is initialized with a unique, non-zero prefix register value. Each prefix value points to a unique page of main storage containing values corresponding to the contents of absolute addresses 0 through 4095 for that CPU.

Prefixing translates real addresses to absolute addresses. For all but real addresses in either the first page of main storage or in the prefix page, the absolute address is the real address. Prefixing interchanges the addresses in the prefixed block with those in absolute storage 0

to 4095. Since each CPU contains a unique prefix register value, each CPU runs with a different and unique set of main storage locations 0 through 4095.

Dynamic Address Translation

Dynamic address translation (DAT) enables a CPU to process with a larger amount of main storage, usually the maximum addressable range of 0 to 16M, than is physically available to the configuration. DAT also enables the CP component in VM/SP with RACF to provide each virtual machine with the maximum range of main storage even if it is not actually available on the current System/370 hardware.

When DAT is in use, addresses refer to virtual storage locations. These reside on external devices in units called pages. Pages are either 2K or 4K bytes in length as determined by the value in bits 8 and 9 of control register 0. For VM/SP with RACF the page size is 4K bytes, the default for System/370 architectures. When a virtual address is referenced, it is first converted, by means of DAT, to a real address, and then, by means of prefixing, to an absolute address.

If the page being referenced by the virtual address is currently in main storage, the (absolute) main storage address within the page is the result of DAT. Only the most recently referenced pages of a virtual storage address space occupy pages of physical main storage. As reference is made to pages in virtual storage that do not appear in main storage, the pages are brought in replacing main storage pages that are less likely to be needed. When the page has been loaded, the absolute storage address within the loaded page is the result of DAT. As program execution proceeds, virtual pages that are in main storage but have not been referenced recently are migrated to their appropriate page on the external device and reloaded when referenced again. Maintaining a separate virtual storage address space for each process provides the isolation between processes.

A two-level table lookup is accomplished to determine whether the virtual storage page referenced by the virtual address is currently in storage and active. The two tables involved in a dynamic address translation are:

Segment Table

The segment table entries are 32 bits in length with the following fields: page table length (4 bits), unused and zero (4 bits), page table origin (21 bits), segment protection bit, common segment bit, and segment-invalid bit.

Page Table

Each page table record entry is 16 bits in length with the following fields: page frame real address component (12 bits), page-invalid bit, extended- storage-address bits (2 bits--not used by VM/SP with RACF, but must be set to zero), and the last bit which is unassigned and not checked during DAT.

The DAT Translation Process

In the following three paragraphs, accompanying figures depict the four steps in DAT. For an illustration depicting the DAT process, see page 21, "Figure 2.2.3".

Dynamic address translation starts with a logical address. Control registers 1 and 0 are consulted to determine the location and size of the segment table to use and the size of pages and segments. The base and displacement are added and the result is broken into three fields, the segment index field, the page index field and the byte index field. A twenty four bit address is used in each case, but the relative size of each field is dependent on a two and three bit field in control register 0.

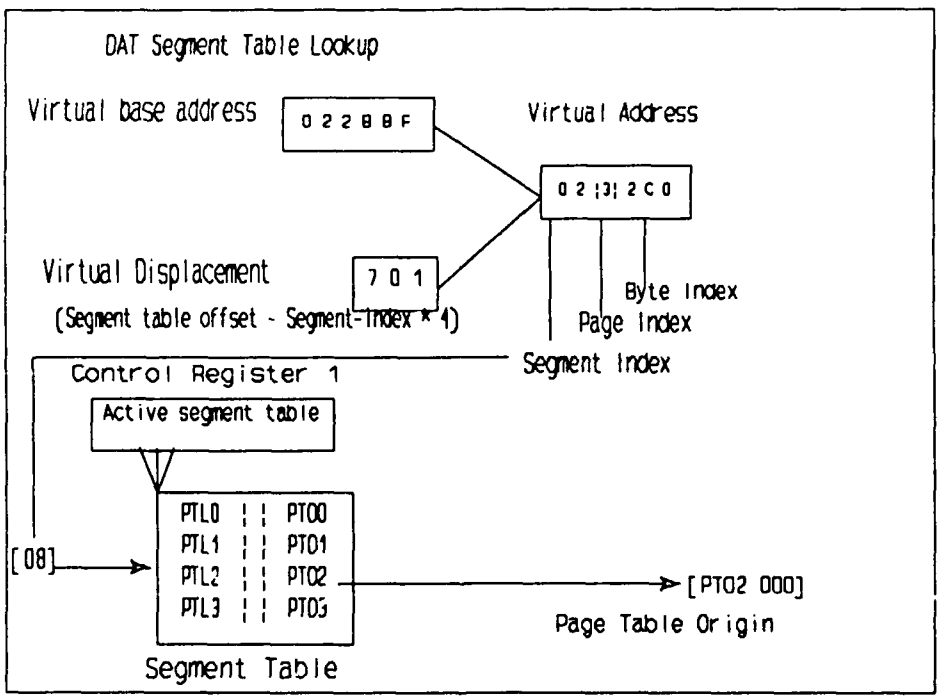


FIGURE 2.2.2

Figure 2.2.2 above describes the segment table lookup component of DAT. The segment-index portion of the virtual address is used to select an entry from the segment table pointed to by control register 1. The segment index is one-fourth of the offset into the segment table. The offset location contains a segment table entry, indicating the page table address to use to find the page table. If no pages are in memory for this segment, there will be no page table, and the segment-invalid bit will be set to one for this segment table entry. The page will have to be retrieved from external storage, a page table built for this segment, and the segment table entry updated. This is performed by VM/SP with RACF.

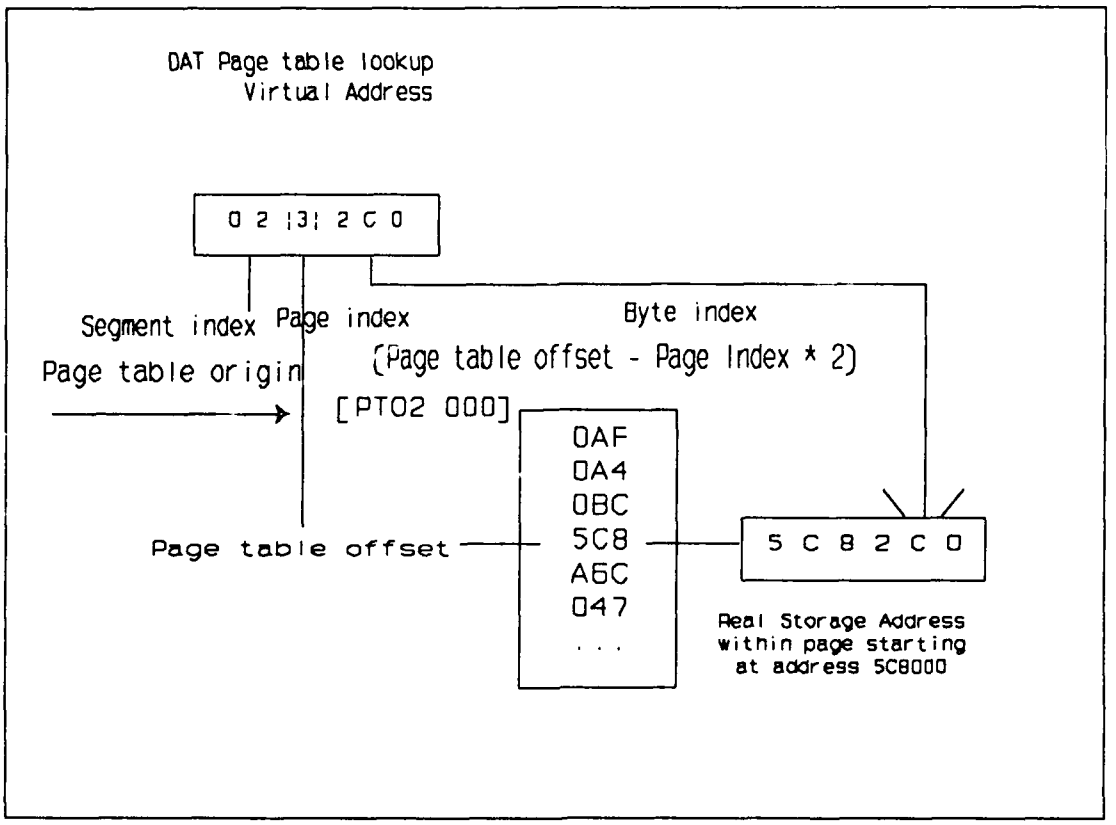


FIGURE 2.2.3

Using the page table and the page-index component of the virtual address, the page table entry is retrieved. This entry contains the upper twelve bits of the 24 bit real address. The lower twelve bits come from the byte-index value of the originating virtual address.

If the page being referenced is not in main storage, a field in the page table entry will be set to one (1). This bit is the page-invalid bit. If it is one, the page will have to be brought in from external storage to complete DAT.

Translation-Lookaside

To speed up the DAT process a translation-lookaside buffer (TLB) is used to enhance performance. Within a short time frame, the sequence of virtual instruction address should refer to the same or a small set of 4K byte pages.

Some of the information specified in the segment and page tables are stored in the lookaside buffer. Prior to invoking the DAT process, the translation-lookaside buffer is searched for an

entry matching the segment table origin in control register 1 and the segment index and page index of the virtual address being translated. If found, the page table information is retrieved and the corresponding real address is constructed from the TLB. If not found in the TLB, DAT is initiated. After DAT, the new real address with its associated segment index and page index is added to the TLB.

Instruction Set

CPU instructions have several formats.

The first byte of an instruction is an operation code. One to five bytes may follow and contain immediate data, references to control information within the CPU, references to information stored in a register, or references to information stored in a main storage location. References to main storage can be made with a displacement contained in the instruction and a reference to a general register. The instruction operand main storage address is found by adding the displacement to the value in the three rightmost bytes of the general register. Some instructions reference main storage with an additional general register, called the index register. The three leftmost bytes of this register are added to the base-displacement address.

All references to main storage may be interpreted as absolute, real or virtual addresses with appropriate address translation applied to derive an absolute address. Specification of the behavior of each instruction is given in [11].

Equivalence within System/370 Architecture

The hardware list in Appendix A indicates several processor types and models that are equivalent in their operation and thus can be used in an evaluated configuration of VM/SP with RACF. The argument that establishes equivalence between any processor in the list follows.

Each of the indicated processors runs native System/370 mode or can be set to run in that mode. Thus the System/370-XA processors are set to System/370 mode and thus operate as described in *System/370 Principles of Operation* [11].

Two processors in System/370 mode are equivalent, according to [11], if any program written for the System/370 mode gives identical results when run on each processor. This equivalence definition also implies that the processors listed in Appendix A present the same abstract machine at the TCB software/hardware interface. Each processor executes the same instruction set. Each of the TCB components will produce identical results on any of the evaluated processors.

IBM has instituted procedures to ensure equivalence across these models and has installed oversight groups throughout the development process to ensure that each software component of VM/SP with RACF gives identical results when run on any of the processors given in Appendix A, and set to System/370 mode.

In particular, this ensures that each TCB component:

(A) has no timing dependencies

(B) does not depend on system facilities being present when the facility is missing from the current configuration

(C) does not depend on system facilities being absent when the facility is in the current configuration

(D) does not depend on results and functions defined to be unpredictable or model-dependent

(E) takes into account changes made to the original System/370 architectural definition as described in Appendix I of [11].

Hardware Support for Multiple Central Processors

In Appendix A, several types of central processors are given: UP, AP, MP, Dyadic, Dual and Partitioned. The case of partitioned, separated configurations is discussed at the start of the hardware section (see page 11, "Summary of System Architecture"). UP processors are processors with a single central processor unit. All other are some form of configuration with two central processor units. VM/SP with RACF will not support more than two CPUs.

AP and Dyadic processors are the same. These configurations consist of two CPUs and a single channel set connected to one of the two CPUs. In this configuration, the control program, CP, for VM/SP with RACF, assigns all input/output through the CPU connected to the channel set. The other CPU will be given main storage bound activities. It is possible to have a channel set path connected to each processor, but only one actively connected to one of the CPUs. If two CPUs have been configured this way and the current channel set fails, the CP can execute the DISCONNECT CHANNEL SET and CONNECT CHANNEL SET instructions to switch I/O processing from the CPU with the failing channel set to the other CPU, adjusting the processing load accordingly.

A dual processor configuration consists of two CPUs directly connected to each other. Each has a directly connected, active channel set with no capability to switch channel sets.

The MP has complete flexibility with regard to channel switching except that no more than one channel set can be connected to a single CPU.

For each of the above multiple CPU configurations, communication can be initiated between the CPUs with a SIGNAL PROCESSOR instruction. This initiates a special interrupt in the

called CPU. Three types of SIGNAL PROCESSOR operations are possible: emergency call which expects a response, external call which does not, and direct call which can allow the calling CPU to control the physical state of the other CPU.

For all the configurations in a system running VM/SP with RACF, only one main storage unit is available. Therefore, only one copy of the dispatcher and scheduler is present in any configuration. Since two CPUs may be present in the configuration, serialization preserving protocols are used. The first involves prefix registers that enable each CPU to deal with unique copies of low main storage. The other form of serialization is accomplished by implementing locks for each of the execution lists and data block lists in VM/SP with RACF. Two phase locking is used to prevent deadlock. See page 283 of reference [42] for a description of this technique.

The COMPARE AND SWAP (CS) and COMPARE DOUBLE AND SWAP (CDS) instructions are used to implement locks which serialize access to counters, flags, lists and control words. During the execution of the COMPARE AND SWAP instruction, no other CPU can perform a store access or interlocked-update access at the main memory location referenced by the CS instruction. This is also true for the COMPARE DOUBLE AND SWAP instruction. In this way, CS and CDS are used to implement semaphores and monitors to serialize access to the single-threaded resources of VM/SP with RACF such as the run list, the timer request queue, the dispatcher stack and the free storage list. (see page 36, "Dispatch and Scheduling")

Hardware Support for Process Management

The CP module executes virtual machines with interrupts handled by CP. Each interrupt gives CP the chance to monitor the instruction sequence of a VM as well as to provide the chance to round robin dispatch at each interrupt.

Hardware Separation and Protection Mechanisms

Several protection features are available with System/370 hardware.

The DAT mechanism provides separation of main memory used by each virtual machine. Through control register 1 the actual segment table to use for translation can be controlled.

Key-controlled protection is available for main storage. Each half-page of main storage can be associated with a storage key value between X'0' and X'F'. The access key of the current PSW must be zero or have a matching value to allow stores in that half-page. A fetch bit is also associated with each half-page of storage to provide for fetch access even when the keys do not match. Key-controlled protection applies to channel-program execution, also, with the subchannel key of the I/O operation providing the access key.

Low-address protection is available on some models to provide additional protection to memory locations 0 through 511. With low-address protection enabled any attempt by a program to store in the effective addresses 0 through 511 (decimal) will not be allowed. Low-address protection is not applied to accesses made by the CPU or channel for such actions

as interrupt handling, updates to the interval timer, and initial-program-loading. Low-address protection is under control of bit 3 of control register 0. When the bit is zero, low-address protection is off. When the bit is one, low-address protection is on.

Segment protection provides a segment-protection bit in each segment table entry. When bit 29 of a segment table entry is one, attempts to store in any virtual address whose (segment table, segment-index) pair maps to that segment table entry will be disallowed, and a protection exception will be raised.

Prefixing provides protection of low memory for each CPU sharing a main storage. A prefix register is available in each CPU indicating the storage section reserved for interrupt PSWs and other low memory values. Prefixing swaps the addresses in the first page of storage with the addresses in the page referenced by the prefix register.

The problem state of execution, which is determined by bit 15 in the current PSW, restricts the instructions that can be executed. This feature was described earlier in this hardware discussion. (see page 14, "Program Status Word")

Hardware Support for Object Reuse

The test block instruction can test for the usability of pages and storage keys in a 4K block. The 4K block is tested for the susceptibility of the block to the occurrence of invalid checking-block code. If the test indicates a usable block, all the 4K bytes in the block are cleared to zeroes. This test is used by CP in initializing blocks of virtual pages at initial reference by a VM to guarantee that they are cleared before use.

Input/Output

The I/O operations involve the transfer of information between main storage and an I/O device. The I/O devices and their control units attach to channels, which control this data transfer.

Channels

A channel connects main storage with control units. It provides direct communication with I/O devices and permits I/O operations to proceed concurrently with non-I/O CPU activities. A channel may be an independent hardware unit, or it may time-share CPU facilities and be physically integrated with the CPU. There are three types of channels: selector, byte-multiplexer, and block-multiplexer.

I/O Devices and Control Units

The I/O devices accessible within a System/370 configuration include card readers, punches, magnetic tape units, direct access storage units, displays, keyboards, printers, teleprocessing devices, communication controllers, and sensor based equipment. These devices may function with an external medium such as paper or magnetically charged surfaces. All I/O device

operations are regulated by a control unit. Logical and buffering functions necessary to operate attached I/O devices are provided through control units.

An I/O operation occurs in one of two modes: burst or byte-multiplex. In burst mode, the I/O device monopolizes the channel and stays logically connected to the channel for the transfer of a burst of information. In byte-multiplex mode the I/O device stays logically connected to the channel for only a short interval of time. A channel capable of operating in byte-multiplex mode can support a number of concurrently operating I/O devices. Data is transferred on the channel from several devices in an interleaved fashion.

The channel facilities necessary to support a single I/O operation are termed a subchannel. As stated above, there are three types of channels. A selector channel has only one subchannel which can only operate in burst mode. A byte-multiplexer channel contains multiple subchannels and can operate in either byte-multiplex or burst mode. A block-multiplexer channel contains multiple subchannels, but can only operate in burst mode. Multiplexing is accomplished between bursts, between command chained blocks and prior to a command retry. A block-multiplexer channel works best with devices that operate in burst mode.

Channel Command Words and Channel Command Programs

Input and output is initiated with either a START I/O instruction or a START I/O FAST instruction. The address of the instruction indicates the channel subchannel and i/o device that this instruction applies to. A channel address word at real address 72 contains the address of a channel command word (CCW). The CCW is the first of one or more CCWs making up a channel program directing the i/o activity.

Once the START I/O is initiated the channel assumes control, and transfer is completed under control of the channel command program. Security considerations for the Channel Command Programs, such as how to prevent self modifying Channel Command Programs, are discussed further in the Software Architecture section on page 44.

Hardware Relevant Initialization

An operating system such as VM/SP with RACF is prepared for continuous operation with a boot sequence known as Initial Program Load (IPL). The primary system device resides on a channel of the system with a specific device address. The load_unit_address control on the console is set to the channel and device address. Then, the load-clear and load-normal keys are activated, in that order.

The load-clear key causes a clear reset to be performed on the configuration. Activating the load-normal key causes an initial CPU reset to be performed on the CPU being IPLed, a CPU reset to be propagated to other CPUs in the configuration, and a subsystem reset to be performed on the remainder of the configuration.

After the resets have been performed, the loading CPU enters the load state. In this state, a channel program read operation is initiated from the channel and I/O device designated by the load-unit-address controls. The read operation is performed in load state as though a START I/O operation were executed but with the channel, subchannel, and I/O device designated by the load-unit-address. An implied channel-address word (CAW) is simulated which initiates the first read into memory location 0 with command chaining. This command chain continues with the CCW which was just read into location 8, as well as any subsequent CCW instructions in the IPL sequence.

When the I/O device signals channel-end status for the last operation of the IPL channel program, the contents of main storage 0 through 7, the first eight bytes read from the load-unit-address, are loaded as the current PSW. Depending upon whether the new PSW indicates EC mode or BC mode, the IPL I/O address is stored in the appropriate low main storage location. If the PSW is loaded successfully, the CPU leaves load state and enters operating state. The loaded program or operating system is in control.

When the CP IPLs a virtual machine, it is this IPL sequence that it is emulating. See page 30, "The Four Major CP Tasks".

Software Architecture

The software component of the VM/SP with RACF trusted system provides separation of the users of the system is such that each user has a virtual machine, each of which acts as if it were a copy of the underlying hardware discussed in the previous section. The Control Program (CP) provides this separation. Other software components provide for trusted administration of the system and controlled access between these virtual machines and the physical storage media, such as tape files that must be read by system tape drives and portions of the system disks. Since CP does not have to handle all the chores that are usually required of an operating system, it is simple both to understand its workings and to explain them.

In the following sections, the software components of the Trusted Computing Base (TCB) are listed, then they are described individually. Following that, the system subjects and objects are identified, and then the protection mechanisms that are used to implement the security policy of the VM/SP with RACF trusted system are reviewed, making reference to the tasks of the individual TCB components.

TCB Component List

CP provides the four major tasks of virtual machine initialization, VM dispatching, memory management, and virtual machine termination. It also provides other services such as translation of virtual channel, controller and device specifications to real channel, controller and device for VM I/O, interprocess communications, spooling to either virtual or real devices and reflection of interrupts to the VM that initiated the actions that caused them.

To perform these tasks, CP must access system tables, including the CP Directory, and interact with the RACF/VM and VMTAPE-MS. The CP Directory contains the information on how to initialize the virtual machine that belongs to each user who has logon information in the Directory. Such information includes the default and maximum allowable size of memory for each user's VM, which minidisks belong to that user's VM, and which minidisks should be linked to this VM at logon time. The concept of resource ownership can be traced to the CP Directory entries, as all the resources that belong to a single user are listed in that user's CP Directory entry. Other resources, such as shared minidisks, which the user has access to but does not own are shown as links in the CP Directory entry[1].

CMS is a single user operating system which runs in most users' VMs. It has a facility to invoke commands such as RACF and DIRM which initiate communication with other components of the TCB, specifically RACFVM and DIRMAINT service virtual machines in the above examples.

The RACF profiles contain the identification and authentication data needed to allow a user to be logged on, plus information on which minidisks and resources each user is allowed to access. The RACF components are used to enter, maintain and backup this information, and interacts with CP to determine what response CP will make to a request from a VM for access to a minidisk or a resource.

Virtual Machine/Directory Maintenance Licensed Program is used by system administrators to enter, maintain and backup the information on the CP Directory. Since the information in the CP Directory and in the RACF profiles database must be consistent, DIRMAINT and RACF both rely on the system administrator to use the DIALOG MANAGER function of ISPF to ensure this consistency. The concept of simultaneously having user information in two system tables, and updating those tables simultaneously, is called dual registration. Since the ISPF/VM runs in a privileged class, and must correctly interpret the instructions to create menus and translate the menu selections to RACF and DMPP instructions, this product must be part of the TCB. Details concerning the function of each of these components are given in the following sections, following a summary list of the components.

---

[1] In addition to the concept of resource ownership maintained by CP, RACF also enforces the concept of profile ownership, such that each profile which describes a user, a group, a connect attribute of a user within a group, or a resource for the purpose of controlling access has an owner. See page 56, "RACF Profiles".

TCB Components

The seven software components of the Trusted Computing Base are

- VM/SP Release 5, including the Control Program (CP) with the Small Programming Enhancement (SPE) as specified in Appendix B. Alternatively, VM/SP High Performance Option Release 5 can be used on hardware that supports this software.

- Conversational Monitor System (CMS) Release 5 is used by system administrators to communicate with Service Virtual Machines such as DIRMAINT, RACF and RACFMAINT. Because of this, it must be trusted to perform its functions correctly when it interprets user commands to the SVMs.

- RACF version 1.8.2, together with the modified version of CMS that supports its interprocess communications facilities.

- Virtual Machine/Directory Maintenance Licensed Program (DIRMAINT) Version 1.4

- VMTAPE-MS Release 4.1, which is a tape management product that, when used with RACF/VM 1.8.2, provides discretionary access control for all tape transactions, and which supports the object reuse requirement for tapes.

- Interactive System Productivity Facility, ISPF Release 2.2, which is used to ensure that users are defined to both CP and RACF in the same way. ISPF must be part of the TCB because its service machine runs in a privileged CP class. (see page B2,"Evaluated Software Components".)

- The ISPF menu descriptors, which are EXECs that are shipped with RACF and which display menu fields which the user fills in at his terminal. The EXEC then issues a list of RACF and DIRMAINT commands to the respective Service Virtual Machines.

In addition, there are a number of stand alone virtual machines which are required for secure operation of the computer system. These virtual machines meet the definition of trusted subject[1] and each is discussed in detail in an appropriate subsection below. They are listed here for easy reference.

- AUTOLOG1 Service Virtual Machine (see page 70, "Subjects")
- AUTOLOG2 Service Virtual Machine (see page 70, "Subjects")
- MAINT Service Virtual Machine (see page 48, "Maintenance Activities")
- OPERATNS Service Virtual Machine (see page 48, "Maintenance Activities")
- RACFVM Service Machine (see page 56, "RACF/VM")
- RACMAINT Service Machine (see page 56, "RACF/VM")
- RACFSMF Service Machine (see page 81, "Audit Mechanisms")
- DIRMAINT Service Machine (see page 65, "DMPP")
- DATAMOVR Service Machine (see page 65, "DMPP")
- VMTAPE Service Machine (see page 67, "VMTAPE-MS")
- ISPVM Service Machine (see page 68, "ISPF") (see page B2, "Evaluated Software Components")

The Four Major CP Tasks

CP acts as a "hypervisor" to provide each VM with virtual disk drives, card readers, card punches, printers, central processing unit, channels, control units and storage as well as real tape drives. Operator functions normally available only to the hardware operator, such as IPL (Initial Program Load), are also provided. To accomplish this, CP owns a special disk volume containing the System Directory, which describes each virtual machine that the system administrator has had defined to it. When a user logs on to the system, CP creates the virtual machine according to the information stored in the System Directory for that user.

Since more than one VM is normally running on the hardware at any one time, CP must partition the real storage so that at least some of the storage of each active virtual machine is always in memory. CP performs memory management using a simple but effective virtual

---

[1] Definition: A Trusted Subject is a subject that runs independently of, or outside of, the underlying operating system of a trusted system, but which has the ability to access or modify security relevant data such as system tables.

storage technique to share the available storage. CP itself manages its own storage by running in real storage, using EC mode but with DAT turned off. See page 14, "Program Status Word". Also, the underlying dispatch algorithm allows each VM currently running to have a share in the processor execution time. When a user logs off, CP is responsible for reclaiming storage and protecting all resources that belonged to the logged off process.

These major functions of CP can best be described as VM initialization, memory management, VM scheduling and dispatch, and VM termination. These functions are described below. In addition, there are descriptions of how CP handles communications between itself and VMs, communications between VMs, translation of device addresses from virtual to real, privilege classes, interrupts, and virtual to real I/O. Also, the High Performance Option software allows CP to make use of hardware features of the top of the line machines in the IBM 370 architecture, and the microcode assists that allow VM/SP with RACF to defer some software processing to firmware, are described at the end of the section.

Before CP can perform its tasks, an Initial Program Load (IPL) must be performed. An IPL is required after power outages, hardware installation and maintenance, and software installation and maintenance. The administrator must shutdown and IPL the new system to activate a change in CP. An operator is required to be present to select the proper options for an IPL at the hardware console and to answer the required VM prompts (i.e COLD/WARM/CKPT startup)[1]. For further information see page 26, "Hardware Relevant Initialization".

During an IPL, VM/SP with RACF will rebuild all control blocks from scratch, thus overwriting the Free Storage Area of memory. One of the default processes of an IPL is to initiate the operator's console and AUTOLOG1 VM service machines. The job of AUTOLOG1 is to log on RACFVM. Once RACFVM is operational it will initiate AUTOLOG2 which has a profile to autolog other needed service machines such as DIRMAINT, DATAMOVR and VMTAPE. So after an IPL all user/service machines start from new. A planned IPL is usually performed after using the FORCE to cause all user/service virtual machines to log out.

Instead of a full IPL, certain conditions can lead to a warm start of the system. This can be chosen by the operator, or it can occur automatically as described below. Whenever a system failure causes an abnormal termination of the real machine that does not result in a disabled WAIT state, VM attempts to reload CP. Often an operator needs to take no action. The system attempts to execute a warm start, thus allowing user's terminals to be reconnected (for logon reinitialization by users) and completed spool files as well as open console spool files to be maintained. In the event of a warm start, device reconfiguration (such as varying a device offline) that was performed by the real computing system operator is remembered by CP for system spooling devices only.

---

[1] Some systems have (or can be fitted with) a hardware/software device to intercept and respond to the hardware (simulate a press of the IPL button) and software (respond to the VM prompts). Such a device is not included in the product under evaluation.

VM Initialization

When a user attempts to log on to the system, CP receives the device interrupt and determines that it is from a device that is not currently attached to any existing VM. The CP routine DMKLOG verifies the userid and authenticates the user from his password. If RACF is running, as is required in a C2 system when users are present, this authentication is handled by the RACF/VM (see page 78, "Identification and Authentication"). Then CP builds a VMBLOK for the user. There is one VMBLOK for each VM on the system, including the service VMs and disconnected VMs that are running without a user logged on to them. There is also a VMBLOK for CP itself.

The VMBLOK is the major control block for a VM. The VMBLOKs are stored in the Free Storage Area (FSA), where they are chained together in a circular linked list. This list contains pointers to other VMBLOKs, scheduling and dispatching information (see page 36, "Dispatch and Scheduling"), save areas for the virtual machine registers and PSW and information about the attributes of the VM, taken from CP Directory, that CP must simulate.

A storage location in the CP nucleus called DMKPSA contains one pointer to the VM onto which the operator is logged, plus a pointer to the system VMBLOK for CP. This latter VMBLOK anchors the chain of VMBLOKs which include all other logged on VMBLOKs. The VMBLOKs themselves are built and maintained in the FSA.

At log on time, CP allocates virtual storage for the VM being created for the user who is logging on, including appropriate segment tables, page tables and swap tables. These tables are also stored in the FSA, although they point into the Dynamic Paging Area (DPA). Virtual channel blocks, virtual control unit blocks and virtual device blocks for the VM are created in the FSA, and pointed to by the VMBLOK, to allow CP to simulate I/O (see page 47, "Printing and Spooling").

Once the VMBLOK is created, CP IPLs (see page 26, "Hardware Relevant Initialization") either an existing system, such as a named system like CMS, or the device address of a device containing a loadable system[1], or else the VM is put into CP READ condition and CP waits for terminal I/O to occur. The entry for the user in the CP Directory determines which of these courses to take. Once the VMBLOK is created the VM can be queued for dispatch.

CP Memory Management

Figure 2.3.1 show the layout of real storage. Page 0 contains Program Status Words for CP, as well as key pointers to system tables. The CP resident nucleus contains CP routines that are so frequently called that it is more efficient to leave them in storage, rather than swapping them in and out. Which routines are included in this area can be chosen when the system is first initialized; there is a default set shipped with the system, and there is no

---

[1] DEF: A Loadable System is one for which the IPL command can be issued and result in the saved system running in the virtual machine that issued the IPL command.

security degradation if a different set of routines is chosen. The Trace area contains a record of recent system events which is kept available for system troubleshooting. Since at C2 there is no requirement for trusted recovery, it is sufficient to say that a privileged command is required to copy this table into a spool file for later reading.

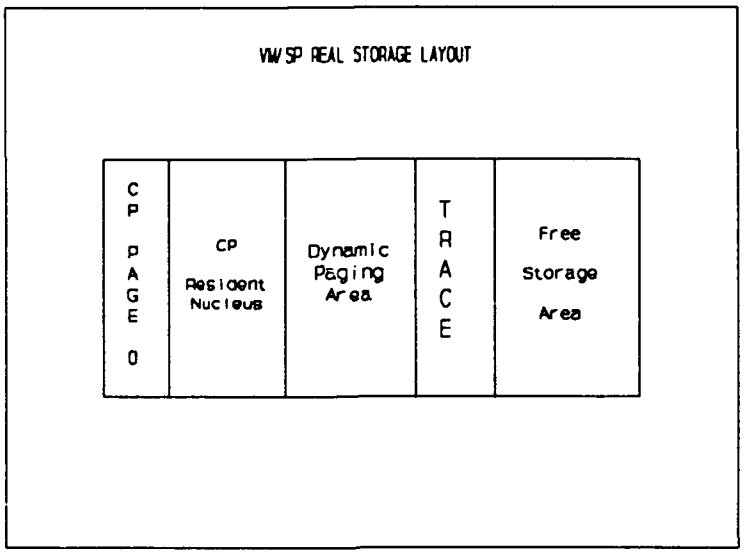


FIGURE 2.3.1

The two areas of storage that are involved in VM processing are the Dynamic Paging Area (DPA) and the Free Storage Area (FSA). The FSA will be discussed in the section on CP's dispatch function (See page 36, "Dispatch and Scheduling"). An example of the type of contents found in the DPA is in Figure 2.3.2.

CP routines that are not kept resident in storage are paged into the DPA, as are pages of the various VMs. The blocks labeled CMS each represent one VM running CMS, although any operating system that will run on the underlying hardware being simulated can also run in the VM. The DCSS is a Discontiguous Saved Segment, which is a public object that can only be written into storage by a system administrator and which can be read or executed by any VM. Its name derives from the fact that when the appropriate command is issued specifying the name of the DCSS, the contents of the DCSS become addressable by the VM at addresses that are greater than, and discontiguous with, the upper limit of the virtual address space of the VM. A DCSS can contain either data or a saved system. CMSNUC is an example of a saved system stored in a DCSS. It contains the common code for the CMS operating system which is being shared by all the VMs that have not modified CMS. The rightmost example, consisting of a VM running CMS, has its own CMS nucleus because the user running in that VM has modified at least one page of the CMS nucleus. This is allowable for any VM; it is not a security issue since such modification can only affect objects to which the user of that VM has been given discretionary access.

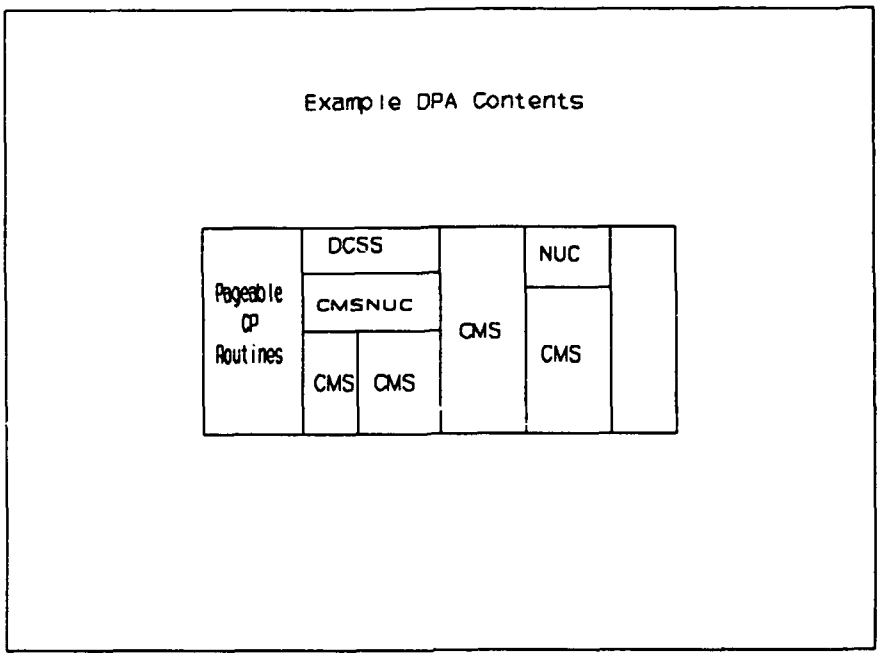


FIGURE 2.3.2

In actual fact, routines are not kept in contiguous storage within the DPA, but are instead swapped in and out using available real memory as the need arises. A paging disk, under the control of CP and not allocatable to any VM, contains the current copy of all pages of each executing VM that are currently not in storage. The swap table for each user points to that VM's corresponding swap pages.

The storage management algorithm use by CP is straightforward. Storage is allocated in 4K byte segments. CP maintains the following system tables to facilitate storage management:

FREELIST
: A list of page frames in the DPA that are used to satisfy a page fault condition. CP attempts to maintain more frames on this list than there are users.

FLUSHLIST
: A list of page frames in the DPA that are used to replenish FREELIST. This list exists only when DPA is overcommitted.

CORTABLE
: A table maintained in the CP resident nucleus showing ownership and status information for each page frame in real storage.

SEGTABLE

For each VM, there is a pointer in the VMBLOK (see page 32, "VM Initialization") to the SEGTABLE in the FSA for that VM. There is one SEGTABLE entry for each 64K of the VM's storage. Each SEGTABLE entry is a pointer to a PAGTABLE.

PAGTABLE

Each page table consists of up to 16 entries, each 2 bytes long. The first 12 bits represent the real page address, and the last 4 bits contains status information about the page, such as whether the page is resident in real storage.

SWPTABLE

For each page table, there is a swap table showing where the copy of each page resides on the system disk. Each page table entry contains the storage key for the first 2K bytes and second 2K bytes of memory, the disk cylinder number and cylinder page number where the page is stored, as well as other system information.

Using the entries of these tables, CP can provide each VM the specified amount of virtual storage needed for that VM's execution, as listed in the CP Directory. CP also uses the same tables and algorithms to swap in non-resident portions of CP. The virtual storage routine is initiated by a page fault, which indicates that a VM attempted to access a word in its virtual memory that was not presently in real storage. A brief description of what happens in response to the page fault interrupt follows:

DMKPTR places VM in the PGWAIT (enabled wait) state. If the needed page is in transit, a request for the page is queued for the dispatcher (see page 36, "Dispatch and Scheduling"). If the page is still in storage (i.e. it is on the FLUSHLIST or FREELIST) and can be reclaimed for this VM, then it is reassigned. Otherwise, DMKPTR obtains a frame from FREELIST and schedules a page in operation if the data currently resides on the pagin disk. Otherwise, this is a first time reference to the virtual page and the real page frame is cleared to binary zeroes. If no page frames are on the freelist, the request for a free page is queued.

Whenever a page is taken from the FREELIST, CP checks that it contains more page frames than there are in-queue users (see page 37, "RUNLIST"). If there are not, then pages are taken from the FLUSHLIST, or the CORTABLE is scanned for unreferenced pages. During the circular scan which starts from the end of the last circular scan of the CORTABLE, the referenced bit is turned off. The bit is turned on again as soon as a VM reads or writes that page, so frequently accessed pages will generally not move from the CORTABLE to the FREELIST. If the next available page frame in storage has been changed, its contents are written to the paging disk before the frame is moved to the FREELIST.

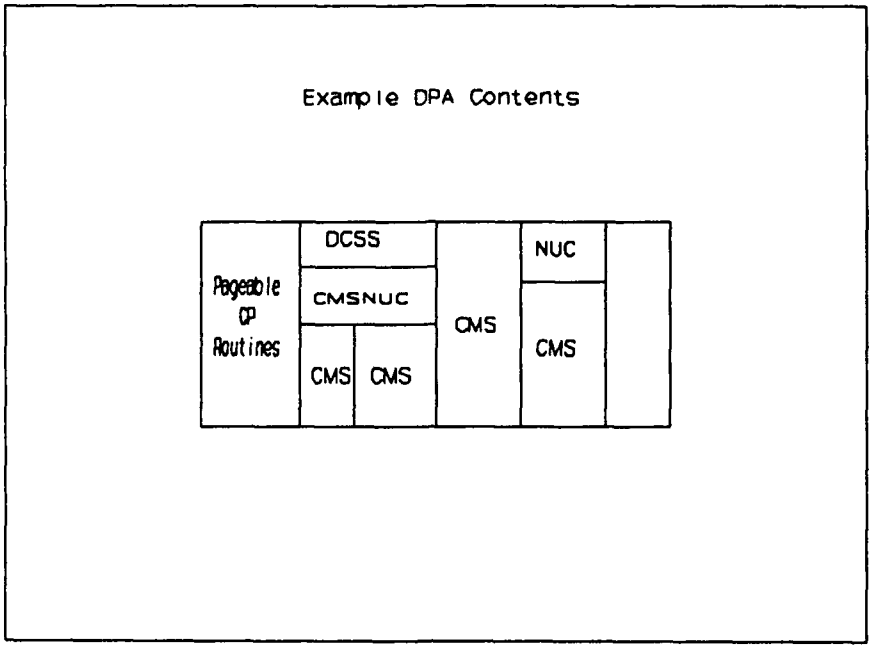


FIGURE 2.3.2

In actual fact, routines are not kept in contiguous storage within the DPA, but are instead swapped in and out using available real memory as the need arises. A paging disk, under the control of CP and not allocatable to any VM, contains the current copy of all pages of each executing VM that are currently not in storage. The swap table for each user points to that VM's corresponding swap pages.

The storage management algorithm use by CP is straightforward. Storage is allocated in 4K byte segments. CP maintains the following system tables to facilitate storage management:

FREELIST
: A list of page frames in the DPA that are used to satisfy a page fault condition. CP attempts to maintain more frames on this list than there are users.

FLUSHLIST
: A list of page frames in the DPA that are used to replenish FREELIST. This list exists only when DPA is overcommitted.

CORTABLE
: A table maintained in the CP resident nucleus showing ownership and status information for each page frame in real storage.

the request queues described above, then it dispatches the VM at the head of the RUNLIST maintained by the scheduler. The functions of the scheduler and dispatcher in maintaining the lists of VMs to run are described next.

The queues of VMs maintained by the scheduler are linked lists of VMBLOKs, with the list headers in locations DMKSCHRL, DMKSCHE1 and DMKSCHE2 in page 0 of the resident portion of CP.

The RUNLIST
is pointed to by DMKSCHRL and contains logged-on users that are currently occupying pages in real storage and contending for CPU cycles. These VMs are termed "in-queue." Each VMBLOK in the RUNLIST is classified as being on one of three subqueues of the list, as follows:

- Q1 - interactive users who are in-queue.
- Q2 - non-interactive users who are in-queue.
- Q3 - in queue users who switch back and forth frequently between Q1 and Q2.

ELIGIBLE LIST
This list contains two queues of VMs that are ready to run but require more storage for their next dispatch than is currently available.

- E1 - VMs that would be considered for Q1 if they were on the RUNLIST. The header for this list is stored in location DMKSCHE1.
- E2 - VMs that would be considered for Q2 or Q3 if they were on the RUNLIST. The header for this list is stored in location DMKSCHE2.

The scheduler maintains these lists in deadline priority sequence within user type[1]. The dispatcher can use the head of the RUNLIST when it next needs a VM to dispatch. When a VM exceeds its deadline priority, or the scheduler determines there is no further work for that VM, it is dropped from all these lists, although its VMBLOK remains active until VM termination (see page 39, "VM Termination").

The VMs on the RUNLIST are further categorized as dispatchable or not, depending on whether or not the VM must wait for an event to complete. The types of wait are PGWAIT, IOWAIT and EXWAIT. EXWAIT means that a CPEXBLOK has been created by another part of CP so that the dispatcher can schedule the work to be done. PGWAIT means that a

[1] DEFINITION - Deadline priority is the time of day by which CP expects a VM to have used its allocated CPU time slice, as determined by an algorithm that has no security implications.

page fault has been initiated for this VM and it is waiting for the paging disk interrupt to signal completion of the swap. IOWAIT means that I/O has been initiated and the VM is waiting for an interrupt from the device to signal completion of I/O.

The scheduler keeps information in the VMBLOK (see page 32, "VM Init.") that categorizes each VM as being in one of 8 possible states. The scheduler maintains each VM state, changing VMBLOK variables as needed to mark the current state of each VM. Figure 2.3.3 lists the state transition table for the various states that each VM can achieve.

State Transition Table for VM states

STATES

1. Q1, dispatchable
2. Q1, non-dispatchable
3. E1
4. no list, wait for terminal action
5. Q2 or Q3, dispatchable
6. Q2 or Q3, non-dispatchable
7. E2
8. no list, active console
   wait or asynchronous wait

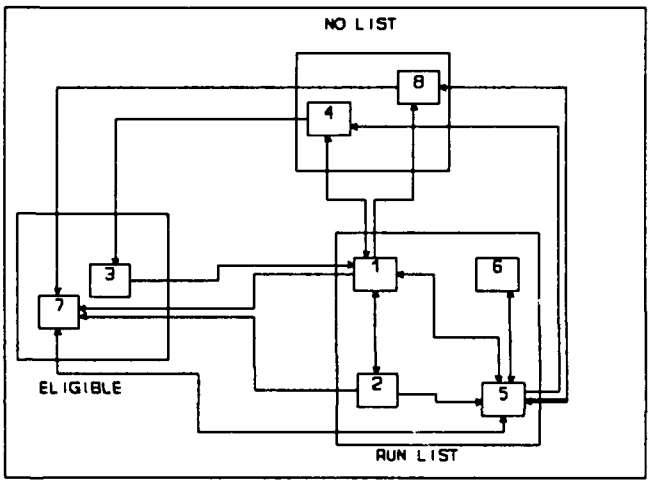


FIGURE 2.3.3

REASON FOR STATE CHANGE

1 --- 2 PGWAIT, IOWAIT or EXWAIT (enabled wait) for any channel
1 --- 4 EXWAIT for terminal read or write
1 --- 5 exceeds in queue time slice
1 --- 7 exceeds in queue time slice, and Q2 is full
1 --- 8 wait without I/O, disabled wait or user hits attention key
2 --- 1 wait condition complete
2 --- 5 or 7 wait completes, but in-queue time slice exceeded
3 --- 1 Q1 now has room
4 --- 1 terminal I/O completes while VM is waiting
4 --- 3 terminal I/O completes, Q1 is full
5 --- 1 terminal I/O completes, VM is active in Q2
5 --- 4 user puts up terminal read or write and enters wait
5 --- 6 PGWAIT, SIO-WAIT or ENABLED wait for busy channel
5 --- 7 dropped for Q2 due to in-queue time slice end
5 --- 8 wait without active I/O, disabled wait
6 --- 5 wait condition completes
7 --- 5 room becomes available in Q2
8 --- 5 or 7 asynchronous I/O, external interruption

VM Termination

When CP encounters the LOGOFF, LOGOUT or LOG command from a terminal connected to a VM, or if the FORCE command is issued by a privileged user, then CP must undo the work done at logon time. This work includes returning all its page frames ir the DPA to the FREELIST, returning all the storage in the FSA for reuse and severing all links to minidisks and any attachments to real devices. If the VM is in any kind of wait state, CP first ensures that the event being waited on completes, but does not allow the VM to continue with any processing that would create another wait state.

Interprocess Communication (IPC)

There are four ways that two VMs can communicate with each other, and all four of them are handled by CP. The four methods are:

- Virtual Machine Communication Facility (VMCF)
- Inter-User Communications Vehicle (IUCV)
- Virtual Channel to Channel Adapter (VCTCA)
- Advanced Program-to-Program Communications (APPC/VM)

Each of the first two types is used by at least one of the trusted service machines in the TCB to perform its work, while the fourth type of IPC is maintained to promote consistency among the various hardware and software lines sold by IBM. IPC is not discussed in the basic C2 Security Guide [28] since there are no special security precautions required of administrators when they set up IPC permissions, or of users when they write or execute programs that use IPC. Correct use of the different forms of IPC is discussed in the references below, and these documents are part of the Trusted Facility Library. Each form of IPC is discussed below, and a description of how the data is buffered is provided for the three types of IPC that provide the data buffering within the VM address space. Since VCTCA emulates a direct connection of hardware channels, any buffering is provided by CP for that type of IPC.

Virtual Machine Communications Facility

In this form of IPC, a VM requests CP to send an external interrupt to another VM telling it that the originating VM has data available. The receiving VM must have earlier authorized VMCF messages, and provided storage within its virtual storage area to place the messages. There is no specific protocol for this form of IPC, but there are general protocols that two communicating VMs can agree to use. These are described below, along with the data buffers used in VMCF.

In a typical application, two VMs that want to communicate each issue the AUTHORIZE control function (DIAGNOSE X'68' with subcode X'0000') and specify the logical address and length of the external interrupt buffer. This must be large enough to accommodate a

fixed size 40 byte message header. Additional control functions, all using DIAGNOSE X'68' with specific subcodes, can UNAUTHORIZE the VM for VMCF, REJECT messages, CANCEL pending messages, IDENTIFY the user's VM as available for VMCF, selectively REJECT messages after inspecting the header information, and QUIESCE and RESUME VMCF operation.

Once a VM has issued the AUTHORIZE command, it can issue SEND, SEND/RECV or SENDX commands, each of which corresponds to a different subcode of the DIAGNOSE X'68' function, to initiate data transfer. When a VM issues one of these commands to CP, along with a specification of the target VM and the VMCPARM parameter list containing the location and length of the message being sent, CP sends an external interrupt code X'4001' to the target machine and stores the VMCPARM data into its external interrupt buffer. Each of these commands specifies a particular data transfer protocol, each of which involves a different arrangement of data transfer buffers. See the *VM/SP System Facilities for Programming* [27] for further details.

SEND
: When a VM issues the SEND command, the VMCPARM specifies to the VMCF handler within CP the location and length of the source data buffer. The VMCF handler issues an external interrupt to the target VM. The target VM issues the RECEIVE command, specifying in the VMCPARM field the location and size of its receive buffer. The VMCF handler then transfers the data from the source VM buffer to the target VM data buffer. Finally, the VMCF handler sends an external interrupt to the source VM so it knows the data transfer is complete.

SEND/RECV
: When a VM issues the SEND/RECV command, its VMCPARM specifies both source data buffer location and length and the location and length of a reply buffer to which the target VM will send a response. The VMCF handler issues the X'4001' external interrupt, the target VM issues RECEIVE with a parameter list that includes the location of the target data buffer. After some amount of processing, the target VM issues the REPLY command with the location of its data buffer in the VMCPARM parameter list. The VMCF handler moves the data from there into the RECV buffer specified by the source VM, then sends the X'4001' external interrupt to let the source VM know that the data exchange is complete.

SENDX
: If a VM issues the SENDX command, the VMCF handler transfers data from the source VM data buffer directly into the external interrupt buffer of the target VM, then issues external interrupt X'4001'. Then the VMCF handler issues the external interrupt to the source VM to let it know that the data transfer has completed. The target VM does not issue either the RECEIVE or REPLY command. The target VM must set aside a large enough external interrupt buffer to handle the largest message that will be sent, and this must be predetermined by the specifications of the communicating programs and must be a parameter of the AUTHORIZE command.

Appropriate return codes are defined for successful and unsuccessful completion of any command. Proper coding technique of the communicating programs should provide handlers for each return code; if they do not, the program will abort. Since any attempt to transfer data out of or into storage that does not belong to one of the communicating VMs results in an error return code, there is no possibility of the VMCF handler in CP violating the separation of VM virtual storage. Since both VMs must AUTHORIZE VMCF communications and set aside memory for the data to be transferred, the discretionary access control requirements are met by this form of IPC.

The Directory Maintenance Program Product, ISPF and VMTAPE-MS use VMCF to establish communication between their service VMs and the VMs of users.

Inter-User Communications Vehicle

IUCV implements the concept of logical paths between two VMs, and has macros specified which carry out the various commands needed in the IUCV protocol. The actual communications with CP involve executing an instruction (X'B2F0') that is not defined in the Principles of Operations [1] but which is interpreted by CP as an IUCV command. Before such a connection is established, the system administrator must have authorized each VM to create IUCV paths with specific VMs or the CP system service (SVM). Allowable entries for a virtual machine X/VM include "IUCV Y" where Y/VM is a specific virtual machine, "IUCV ANY" which allows X/VM to send a CONNECT command to any named VM, and "IUCV ALLOW" which means that any VM can send a CONNECT to X/VM and X/VM will see the interrupt.

Each VM that is using IUCV must declare two internal buffers for outgoing and incoming messages. In actuality, these buffers may be allocated before or after the connection is established, and their size may be dynamically determined. A VM's buffers may coincide or overlap. If the sending VM sends a longer message than the receiving VM has room for, CP will break the message up into smaller messages that fit the receiving buffer. Messages of arbitrary length can be sent in either one way or two way mode. Since both VMs must take a specific action to enable IUCV communications, the discretionary access control requirements are met by this form of IPC.

Each IUCV command can be programmed by using the IUCV macro in Assembler Language, or it can use support macros in CMS which can be invoked from Assembler Language when the program is running in a CMS environment. The external interrupt used is X'4000', and all IUCV communications involve a fixed size (40 byte) external interrupt buffer that is declared with the DECLARE BUFFER command. See the *VM/SP System Facilities for Programming* [27] for further details.

A typical two way IUCV session between X/VM and Y/VM illustrates the use of IUCV. In the following, assume both X/VM and Y/VM have each set aside a SEND and RECEIVE area of some (not necessarily equal) size:

1. Both X/VM and Y/VM issue the DECLARE BUFFER command with the address of its own external interrupt buffer.

2. X/VM issues "CONNECT Y" to establish communication with Y/VM. The IUCV handler within CP checks if this connection is authorized by the CP Directory entry, and if it is, sends an external interrupt for Y/VM.

3. Y/VM gets the external interrupt and inspects the information in the buffer.

4. Y/VM issues the ACCEPT command, and the IUCV handler sends an external interrupt to X/VM indicating a completed connection.

5. X/VM gets the external interrupt in its buffer. Normally, X/VM has created an interrupt handler for this case so that it could have been doing other processing since it issued the CONNECT command. It might also have gone into a busy wait until it got this interrupt.

6. X/VM issues the SEND command, which does not send data, but queues an external interrupt to Y/VM and put information in the buffer that X/VM has a message ready to send. One parameter of the SEND command specifies whether the message is one way or two way.

7. Y/VM either gets the external interrupt and it is reflected to the interrupt handler, or else Y/VM periodically issues the DESCRIBE command to see if there are any pending IUCV messages.

8. Y/VM issues RECEIVE, which includes the location and length of the receive buffer. The IUCV handler knows whether it is receiving a one way or two way message. If it is one way, then it sends X/VM an external interrupt indicating completion after all data has been transferred from the buffer in X/VM to that in Y/VM. Since the buffers are not necessarily the same size, Y/VM may have to issue multiple RECEIVEcommands until all data is transferred. CP will issue the external interrupt required for each RECEIVE.

9. For two way messages, X/VM expects a reply. Y/VM forms a reply in its send buffer and issues the REPLY command. The IUCV handler issues an external interrupt to X/VM indicating that a reply message is ready. Y/VM continues processing without waiting for any indication that X/VM has gotten the message.

10. X/VM learns that its message has completed either from the external interrupt or, if external interrupts are turned off then X/VM can periodically issue the TEST COMPLETION command to see if the message has been transferred. The TEST COMPLETION return code will reflect whether the transfer has completed.

11. X/VM issues the SEVER command to close the connection. The IUCV handler sends an external interrupt to Y/VM to indicate that the connection has been closed.

12. Y/VM gets the external interrupt and reflects it to an interrupt handler, which does any clean up needed.

13. Y/VM issues SEVER, which lets the IUCV handler know that both VMs are done. Then the IUCV handler can close the connection and clean up its control block.

14. Both X/VM and Y/VM issue RETRIEVE BUFFER commands to recover the 40 byte external interrupt buffer.

Appropriate return codes and program interruptions are issued by the IUCV handler. Among these are codes corresponding to attempts to read or write to virtual storage outside the range of one of the connected VMs.

Users who issue RACF commands from within their VMs have their requests handled by CP, which communicates with RACF/VM and RACFMAINT/VM through IUCV messages. Each of these machines has "IUCV ANY" in its CP Directory entry. (RACFSMF/VM does not communicate with users, so it has no corresponding statements in its CP Directory entry).

Virtual Channel-to-Channel Adapter

CP can also simulate a connection that is used in real machines to link an I/O channel from one machine to that of another. One must declare a virtual channel, using the CP DEFINE command, then issue the CP COUPLE command, specifying both the target VM's userid and virtual channel designator. The target VM must do the same. Then the individual VMs can use regular hardware I/O commands such as SIO, and receive and handle interrupts reflected to VM by CP, as if it were really connected by a physical cable. Due to the requirement that both VMs must DEFINE and COUPLE to a virtual channel, and this channel must be specified by both the userid and channel designator corresponding to the target VM, the discretionary access control requirements are met by this form of IPC.

Advanced Program to Program Communication

APPC/VM is the VM implementation of the IBM SNA protocol. When used in a single VM/SP with RACF system, it simulates communications between two machines on a network. It provides half-duplex communications between any two VMs. Both VMs must set aside a buffer for messages, and the receiving VM must have authorized the receipt of APPC messages. The VM/SP implementation of APPC uses the same interrupt causing instruction as IUCV. Two communicating VMs would use the same communication protocol as the one described above for IUCV. APPC/VM is invoked, rather than IUCV, by using the APPCVM macros to issue the DECLARE BUFFER, CONNECT, SEND, RECEIVE, or SEVER command (two way messages are not supported, so REPLY is not used). One essential difference between IUCV and APPC/VM is that a user VM may connect to another VM, but also can connect with a resource through its server. For this to happen, the TSAF SVM must be running (see page B-2, "Non-TCB Software Comp.") and the resource must be identified to TSAF by using the *IDENT system service. When TSAF/VM is started, it notifies CP of the names of all its available resources, each of which can be a program, data file, a device or any other entity that can be accessed through a server. Only local resources

can be identified in the evaluated configuration, while global resources can be connected to by a TSAF SVM connected to a network (TSAF collection) of VM/SP systems. See *VM/SP Transparent Services Access Facility Reference* [26] for more details.

The following list of differences between similarly named commands is helpful in understanding how APPC/VM differs from IUCV:

CONNECT
The source VM can issue the name of a resource, rather than that of a userid.

RECEIVE
The VM must provide a path ID (returned by the APPC/VM handler when CONNECT is issued) to specify which communications path you are receiving on; the RECEIVE may be issued before any data has arrived on the path; and the WAIT=YES option allows the program to wait for data.

SEND
One can use either the IUCV SEND or the APPC/VM SENDDATA command to notify the recipient that you have data to send. The parameter lists and available options are different.

SEVER
In APPC/VM, the user data field does not exist, and there are both NORMAL and ABEND types of SEVER.

APPC/VM only functions
APPC/VM has additional protocol commands: SENDCNF, SENDCNFD, SENDERR, SENDREQ. These commands do not affect the way buffers are used by APPC/VM, but enhance half duplex communications.

IUCV only functions
IUCV has protocol commands that are not implemented by APPC/VM: PURGE, QUIESCE, RESUME, REJECT, and REPLY. These commands do not affect the way buffers are used, but enhance full duplex communications.

Because both the user VM and the server VM must agree to this form of IPC by issuing the CONNECT and ACCEPT commands, the discretionary access control requirements are met by this form of IPC.

Real and Virtual I/O

CP performs I/O to real devices. Unlike most operating systems, however, it does not provide device drivers, channel programs or other familiar services. Instead, when a VM's resident operating system writes a channel control program and performs the instruction to initiate I/O, CP refers to the VMBLOK for that VM and translates from virtual channel, controller and device numbers to the real channel, controller and device numbers that correspond to that I/O device. CP also handles its own paging I/O, and the spooling of files to real printers (see

page 47, "Printing and Spooling"). CP must also handle real I/O to devices such as the console and tape drives.

To perform all these real I/O functions, CP uses channel programs, often referred to as SIO/SIOF from the machine instruction that initiates the I/O. VM I/O consists of SIO/SIOF to a virtual console, SIO/SIOF to a non-console device, SIO/SIOF to a spooled device, and I/O request via the DIAGNOSE instruction. CP simulates the SIO/SIOF instructions by maintaining virtual channel, control unit and device blocks which it maps internally to real devices which are attached to actual channels and control units. This mapping is set up when any of the CP commands ATTACH, DEFINE or LINK is issued by the VM, and maintained by CP using IOBLOKs in the FSA, so that no user can access or modify a real device without CP intervention.

The SIO/SIOF instruction creates a communication path between main storage and a device. The device is specified using a 16-bit address, where the first 8 bits specifies a channel, and the 8 low order bits identify the device and control unit on that channel. A VM user, usually through the operating system running on his VM, must first create a channel program consisting of Channel Command Words (CCW) which specify what data is to be transferred. Then the base address of this channel program is put into the Channel Address Word (CAW). Then the SIO or SIOF instruction is executed, and information about the I/O operation is stored in the Channel Status Word (CSW). When the SIO or SIOF instruction is executed, the CSW is sent to the channel device, which then executes the channel command program (a string of CCWs with the chain bit set on all but the last CCW in the sequence) that it finds in main storage. The SIO instruction holds up CPU processing until the first instruction of the channel program has completed, whereas the SIOF command does not wait. Both these instructions are privileged, so when they are issued by a VM they are trapped by CP and simulated. (See page 47, "Real and Virtual Interrupts".)

Self modifying Channel Command Programs are prevented by CP, since prior to allowing the I/O to proceed, CP intervenes and makes two modifications to the CCWs to ensure that system security is not compromised during execution of a channel command program. CP checks that all intended disk operations will be to and from minidisks accessible by the VM machine, and CP builds a real channel command program from the VM's virtual channel command program and transfers the real channel command program to a read-only section of CP's real memory prior to initiating the I/O. During this process, CP compares all CCW commands to a list in its memory of CCW commands that are supported by CP. If an unsupported command is found, then CP still completes its translation of the Channel Command Program but only allows the program to execute as far as the unsupported instruction. CP then returns the error code from the channel device to the VM. As a final consideration of the security of channel command programs, it is not possible for such a program to modify the translated channel command program in CP storage because the CCW commands are neither self referencing nor can any supported CCW commands reference the CAW. The CAW acts as the program counter for the channel command program and is passed to the channel device at the beginning of SIO or SIOF execution. Only the channel device can read or modify the CAW. Thus, a channel command program could not write data over itself, and thereby modify itself, because it could not tell the channel device where in real storage to write the new code.

CP sets up an IOBLOK for each device in a VM's I/O configuration. These blocks are linked together in the FSA and pointed to by a field in the VMBLOK. While CP maintains a list of virtual channels and virtual control units, these do not correspond to real channels and control units. However, use of the DEDICATE statement in the CP Directory or use of the ATTACH command does create a one to one correspondence between a virtual device and a real device, whether it is a tape, disk or other discrete I/O device. The LINK command creates several virtual device blocks for a single real device since several VMs are sharing a single minidisk or other device. The IOBLOKs for each virtual machine are pointed to by a field in the VMBLOK, (see page 32, "VM Initialization"). These blocks are maintained in the FSA by CP and are not addressable by any VM. Thus, CP mediates all I/O by either simulating the SIO/SIOF command when a virtual device, such as a spool file, is addressed, or else translates the SIO/SIOF command to a real device address before running the I/O program.

VMs may also issue Diagnose commands that CP interprets as requests for I/O. These Diagnose commands are:

DIAGNOSE X'18'
: performs simple disk I/O using a standardized channel program format.

DIAGNOSE X'20'
: performs I/O to tapes, disks or unit record devices dedicated to the virtual machine. VMTAPE-MS mediates the ATTACH command to maintain discretionary access control over tape files.

DIAGNOSE X'58'
: performs console communications with a virtual 3215 terminal. This allows the virtual 3215 terminal to be supported on a real 3270 terminal.

DIAGNOSE X'98'
: allows a VM to lock and unlock virtual storage pages into real storage, and execute channel programs that do not need CCW translation by CP. Use of the DIAGNOSE 98 command is a use of privilege, and must be restricted accordingly. The Trusted Facility Manual document [28] cautions the system administrator to audit the use of DIAGNOSE X'98' and not to place the DIAGNOSE X'98' privilege in the OPTION statement of any user's CP Directory entry.

In summary, CP and each VM use the same IBM 370 instructions to perform I/O. However, when a VM issues such an instruction, CP interprets it and performs the corresponding I/O to a real or virtual device, as appropriate.

Printing and Spooling

CP provides both real spooling[1] to control the movement of files between disk and unit record devices, and virtual spooling to control VM access to virtual devices, such as the virtual card reader, virtual card punch, and virtual printer. Although virtual spooling has many of the characteristics of IPC, it is handled by the same CP software that handles real spool files to real devices. For this reason, its use and control is discussed as I/O, rather than as IPC. There are CP commands available to enable the contents of the virtual printer to be spooled to a real printer. The virtual card reader and punch are used to provide input to and receive output from programs and utilities run by the VM, and to send messages to users on other VMs.

CP handles both real and virtual spooling in a similar way. Space is allocated in the DPA for a buffer of an appropriate size, and then data is read from either a real device or a VMs virtual device through the buffer into a spool file reserved in CP's own disk, SYSRES. Similarly, a buffer is created in the DPA and used to transfer data from a spool file through a buffer in DPA either to the specified real device, or to a user's virtual device. The necessary control blocks and their formats are described in the design documentation for CP [16].

Real and Virtual Interrupts

Interrupts can be caused by or for each virtual machine running under CP control or can be caused by or for CP itself. CP controls all system resources so that its operations are transparent to each virtual machine. Thus, when an interrupt is caused by or for a VM, CP reflects the interrupt to the virtual machine by storing the information concerning the interrupt in the VM's page 0 in the same locations that would have been used if the VM were running directly on the IBM 370 hardware. When the VM is next dispatched, it appears to that VM as if it had just had an interrupt.

When a real interrupt occurs, (see page 15, "Interrupt Processing") a first level interrupt handler gets control of the CPU, saving the status of the system, decoding the interrupt and processing it, then returning to the dispatcher. CP has handlers for five major types of interrupt:

- External
- SuperVisor Call (SVC)
- Program
- I/O
- Machine

---

[1] DEFINITION: SPOOL - Simultaneous Peripheral Operations OnLine

There are four possible external interrupts. The interval timer is set to the maximum time a VM is allowed to run. If a VM runs for that long with no interrupt of any other kind, then the interval timer causes CP to reschedule the VM. If the operator is logged on at a console and hits the interrupt key he is disconnected and can log on at another device. The CPU timer interrupts when a VM has run its maximum time slice, and is rescheduled. The clock comparator interrupts when the time reaches that specified in the TRQBLOK on top of the

stack of timer requests. The TRQBLOK is placed on the dispatch queue along with the IOBLOK requests so the dispatcher can perform the task requested.

When a SVC is detected, (see page 22, "Instruction Set") the response depends on whether CP was executing in real supervisor state, or real problem state. If CP was in problem state then it was executing a VM, so the interrupt is reflected back to the VM that caused it. If that VM's page 0 is not in storage, a page fault occurs and the memory manager handles the resulting situation. If CP is in real supervisor state, then the SVC is handled accordingly.

A program interrupt results from normal paging requests, attempts to execute privileged instructions, and program errors. If CP was executing in real supervisor state and the event was not X'40' Monitor Event, which causes the event to be recorded and then returns to the dispatcher, then the system is re-IPLed. If CP was in problem state, then CP handles the page fault if that was what caused the interrupt, or else it reflects the interrupt back to the VM that caused it. If the VM was executing in supervisor state, CP simulates the instruction appropriately.

Completed real I/O operations result in an I/O interrupt. The I/O interrupt handling routine checks the queue of IOBLOK requests and TRQBLOK requests for the appropriate I/O request, and signals ending status to the virtual machine or the CP module that caused the IOBLOK to be built.

A machine malfunction causes a machine check (MCH) interrupt, and the handler attempts to recover from the malfunction. If the problem is corrected, then the event is logged to an error recording cylinder of SYSRES. If the MCH handler fails to recover from the error, then the interrupt code is stored in the fixed logout area of storage and processing continues, if possible.

Maintenance Activities

CP also has capabilities to handle many hardware and software maintenance activities. For example, anyone with class C CP privilege can invoke the CPTRAP command, which will create a spool file in the privileged user's RDR list that contains information from the TRACE area of real storage. See page 32, "CP Memory Management" for details of the TRACE area. This area keeps a circular list of all commands issued by all VMs on the system, and can be used to determine the source of a system problem. Any user with CP privilege class C can specify any of X'1B' different types of events to be extracted from the TRACE area and spooled to a RDR device. Although the default device is the RDR of the invoking privileged user, any other user's RDR can be specified. For example,

CPTRAP 1 2 3 X'1B' TO MAINT

would spool all events that involve external interrupts (type 1), SVC interrupts (type 2), program interrupts (type 3) and clear I/O instructions (type X'1B'). The results would be spooled to MAINT, since that is specified in the command. If the destination had been left off, then it would have been spooled to the invoking user's RDR device. Alternatively, any user with CP privilege class E can inspect real storage. If that user knows where the TRACE area is located, the information stored there can be read and processed. Again, a CP privilege is required to do this. Errors in individual VMs can be diagnosed by executing the VMDUMP CP command, which copies only the specified range of virtual memory belonging to the user who invokes the VMDUMP to the specified RDR device. The default is the user's own RDR, but the spool file can be sent to anyone. The IPCS program, which is supplied with VM/SP with RACF, can be used to inspect the storage contents.

Both of the above methods allow a static inspection of real or virtual storage. CP also has commands that allow any class G user to interactively inspect an executing program. For example,

CP ADSTOP 3000

sets a dynamic breakpoint at virtual address X'3000', after which a VMDUMP, DISPLAY or DUMP command can be used to send a storage area to the RDR, display storage at the terminal, or send the value of all registers, PSW, and specified storage areas to the PTR spool file, respectively. BEGIN will continue execution of the program.

There is also a debugging program that is more easily usable by any class G user, since it not only displays the hexadecimal contents of storage, but translates instructions to their mnemonic form, checks storage locations for actual changes in contents, and otherwise provides a dynamic debugging option. PER is a standard CP command and only runs in a user's VM.

There are also two Service Virtual Machines that a privileged user can log onto. MAINT has all CP privilege classes, A through G, and runs utilities that allow read and write access to all data on the system. OPERATNS normally runs without privilege, but its RDR is the default target of all automatic DUMPs that are caused by program ABENDs. Since dumps can contain sensitive information, OPERATNS must work correctly. Its most usual use is to run dump analysis programs such as IPCS.

High Performance Option

A software enhancement upgrade called the High Performance Option (HPO) is available to install as part of VM/SP with RACF. This feature is a software component whose primary feature is performance improvements for VM/SP with RACF, as could also be done by upgrading the current processor. HPO is not installed as a separate section of code, but rather as updates throughout the CP component of VM/SP with RACF to allow it to get performance gains normally only available with hardware features typical of larger System/370 machines.

The performance enhancement support provided by HPO is internal to the TCB. Except for the change requiring spool files to be identified by both spoolid and userid, the user interface does not change with installation of HPO. Security relevant features of HPO are discussed in the C2 Security Guide [28] and the HPO Installation Guide [36] is part of the Trusted Facility Library.

The features of HPO are outlined below.

Extended Storage Support

High-end System/370 processors have the capability of supporting main storage in excess of 16,777,216 bytes (16M). This feature, called extended real addressing, provides up to four (4) times the main storage of systems without this feature. Each VM machine being supported by VM/SP with RACF still runs with the 16M main storage limit. However, CP is able to manage all available storage and ensurea smaller virtual to real storage ratio for the whole VM system.

Extended Channel Support

HPO supports the high-end processors that allow more than 16 I/O channels to be in a channel set. Up to 32 channels can be connected. This requires four digit device addresses. It does not extend virtual device addresses.

Enhanced Paging Device Support

Certain models support an external high-speed paging device capable of moving 4K pages into and out of real storage very rapidly. HPO uses this device as another paging and swapping device, even though it is not really an I/O device.

Vector Facility Support

The vector facility is an enhanced set of instructions permitting faster scientific and engineering computations. With HPO this feature is made visible to the VM machines running under VM/SP with RACF. It adds additional registers, but no communications capabilities. It gives appropriate interrupts to Vector Facility instructions executed on VM machines.

MP Support

Separate dispatcher queues have been established for each processor in an MP configuration. Several locks have been split into multiple finer-grained locks, increasing concurrent processing capabilities. When idle, each processor initiates an "active" wait, looking for work to be added to its queue rather than waiting for interrupts from the other CPU. Free storage is kept in processor-local free storage subpools. Storage freed by a processor is placed on its local free storage list. This list is searched first before interrogating the list of the other processor. HPO provides enhanced page migration and swap table migration support for MP systems. The granularity of reference is the page not the segment, allowing HPO to identify unreferenced pages to move even if another page within the segment has been recently moved. Swap migration is handled separately from page migration in HPO permitting a 16 page segment to be migrated even if the pages in the segment haven't been migrated.

Spool File Limit Relief

In VM/SP with RACF there is a system wide limit of 9900 active spool files. HPO permits each user to have up to 9900 active spool files. This does impact the user interface. A spool file in HPO is addressed by a doublet containing the userid and spoolid. This doublet changes if a file is transferred to another user. With HPO, spool files can be dumped to multiple tape volumes. The major change internally for spooling is that the spool file queue for reader files was moved from real storage to virtual storage. Processing spool queues by privileged users must be done by using Diagnose X'D8' rather than following real storage pointers.

Extended Virtual Device Limit

The number of virtual devices that a single virtual machine can communicate with has been increased by a factor of eight.

Shared Page Protection with Segment Protect

Segment protection is a System/370 hardware function which HPO uses to ensure that shared pages are not altered.

Cached DASD Support

Four 3880 DASD devices support caching of either 4K page data (models 11 and 21) or arbitrary user data on minidisks (models 13 and 23). Minidisks in the CP directory may be defined to be cached or not cached by the 3880.

No Extended Architecture Features

HPO is a performance, not a feature, enhancement option. Except for those noted above, no XA features are simulated by HPO. HPO does not provide support for the System/370 Extended Architecture. Notice that in order for VM/SP with RACF to execute on a System/370-XA processor, that processor must be set to System/370 mode.

Microcode Assists

During installation of the system, the system administrator may choose to have certain instructions, which are normally simulated by code in CP, be performed by microcode. Then when either CP or a VM running under VM/SP with RACF attempts to execute one of these instructions, it is deflected to the firmware rather than being simulated by code within CP. The system administrator determines which microcode assists will be handled by issuing a CP command (SET ASSIST ON). This sets appropriate bits in Control Register 6 to indicate to the system which of the six microcode assists is active. Individual assisted functions can also be activated or inhibited by proper setting of bits in Control Register 6. Each type of assist is discussed below.

Virtual Machine Assist

A virtual machine assist is used to enhance the performance of a program executing in virtual Supervisor mode in a Virtual Machine by directly executing any of the 12 instructions: Insert PSW Key, Insert Storage Key, Load PSW, Load Real Address,

Reset Reference Bit, Set PSW From Address, Set Storage Key, Set System Mask, Store Control, Store Then AND System Mask, Store Then OR System Mask, Supervisor Call (SVC). The firmware will be used to perform the instruction, rather than simulating it in the CP software, as long as certain conditions are met that allow the CP simulation to be bypassed. If any necessary conditions are not met, then these instructions are simulated by CP instead.

Control Program Assist
The Control Program Assist is used to accelerate completion of various parts of CP. It provides optimized code for 22 commonly performed CP tasks.

Expanded Virtual Machine Assist
The Expanded Virtual Machine Assist may be used on a system on which both the Virtual Machine Assist and the Control Program Assist have been set on. It provides complete or partial direct execution of 11 virtual machine instructions.

Virtual Interval Timer Assist
This assist emulates the real hardware timer, maintaining a virtual machine interval timer value for each VM and causing virtual or real program interruption when the interval timer is decremented through zero.

Virtual Machine Extended Facility Assist
The Virtual Machine Extended Facility Assist allows a virtual machine to directly execute 12 instructions of the System/370 extended facility, provided that facility is installed.

Shadow Table Bypass Assist
This assist is not used in the evaluated configuration since it improves the speed of machines using the Virtual=Real option. This option is not part of the evaluated configuration.

None of the five allowable assists changes the way any component of the TCB works. All TCB code remains unchanged whether the microcode assists have been turned on or not, and the path of execution also remains unchanged to the extent that the assisted instructions are either executed directly by the firmware or simulated by code in CP. For this reason, no security relevance is attached to whether any of the five assists have been turned on for a particular system configuration.

CMS

The Conversational Monitor System is a single user, fixed memory operating system that runs in a Virtual Machine created and maintained by CP. Because CP handles all scheduling and dispatching chores, performs all memory management tasks, and translates I/O requests to devices that CP attaches to the VM on request, CMS need only deal with a single user with a fixed address space and a small set of I/O devices. Thus, its tasks are reduced to the following:

- Initialization
- Command interpretation
- Program management
- Fixed storage management
- Input/Output for attached virtual I/O devices

Most of these are accomplished by sending a DIAGNOSE command to CP. For purposes of describing how CMS works, Figure 2.4.1 shows how CMS views memory layout for its virtual machine. Fixed addresses are given in their hexadecimal notation, while addresses that can either be set at system installation time, or depend on dynamic operation of CMS, are shown as the name of the system variable in which the address is stored. All pages in each region are protected by a memory protection key (See page 14, "Program Status Word"); the value of the access field of the storage key assigned to each page in a region is shown in the lower right hand corner.

Initialization

The working copy of CMS is created when the VM/SP system is first generated from the load tapes. The Installation Guide provides details on how to create a named discontiguous saved system (DCSS) during the initial generation. The location of the CMS DCSS on the system disk SYSRES is maintained in module DMKSNT so that when a user issues the command 'IPL CMS', or when his default VM as described in the CP Directory lists CMS as the default system, the segment table entries corresponding to the CMS shared segments are loaded into the segment table pointed to by the user's VMBLOK. In addition, segment table entries for the remainder of the user specific parts of memory are created and appended to the segment table.

The CP Directory entry for the user also has information about the mini disks that belong to the user. Pointers to the virtual channel, controller and device blocks for these mini disks are added to the user's VMBLOK, as are pointers to description blocks for the console device and the default PUN, RDR and PRT devices. After adding these control blocks, the system is ready for command interpretation.

Command Interpretation

Since CMS is a single user operating system, a simple tokenizer is used to collect each command and its corresponding arguments. A data structure called a "tokenized P-list" is used to pass this information on to be interpreted as a command. The command interpreter checks whether the command is one of an EXEC, a nucleus extension, a program already loaded in the transient area, a program in the nucleus, an executable module on the disk but not yet in the transient area, or a CP command, in that order. This exhausts the possibilities of commands that can be entered at the terminal when in a CMS session. Each is described below.

EXEC
: An EXEC is a file containing commands for one of three independent command interpreter systems: REXX, EXEC, or EXEC2. EXECs are usually distinguished by a file extension (file type) of EXEC. These EXECs may reside on any of the minidisks attached to the user's VM and can be invoked with a single call. CMS scans all the attached minidisks in the Active Disk Table for a file of type EXEC with the name of the command, then if it doesn't find it, it searches all the minidisks for a file with no extension with the same name.

nucleus extension
: A program to be executed in the CMS virtual machine is normally loaded into the transient area when it is called, overlaying any previous program there. It is possible for the user to load a program into the lowest order DMSFREE area, if there is room, and remain there for the remainder of the session. This can be done for frequently used programs, such as connecting to a Data Base Management Virtual Machine.

Transient area program
: A user application is normally loaded into the transient area with the LOAD command. If the program is already there, the CMS command interpreter immediately transfers control to the entry point of the program there.

Nucleus Program
: Certain commands are run by invoking programs in the nucleus. If the command interpreter encounters one, control is transferred to the entry point of that program in the nucleus area.

Disk Module
: The output of the load process can create load modules on a minidisk. If the command interpreter encounters one of these, it is transferred into the transient area and control is transferred to its entry point.

CP command
: If nothing else matches the command, the CMS command interpreter passes the command and its arguments on to CP to see if it can interpret it.

Program Management

User programs are created by compiling and loading source text. Since each Virtual Machine has the same memory layout within the same VM/SP with RACF system, any user can create a load module by running one of the loader programs on the output of a compiled program and share that program with any other user by allowing a link to the minidisk that contains it. The possible loader commands are LOAD, GENMOD, and NUCXLOAD. The first creates an executable version of the compiler output it was given as an argument and places it in the transient area. The second creates a loadable module on disk, which will be found by the CMS command interpreter when its name is invoked. The third takes the name of a loadable module created by GENMOD and creates a nucleus extension, which will be found by the CMS command interpreter when its name is invoked. Since only one program can be run at a time in each VM, the program runs until completion, error, or the program is stopped from the terminal.

Fixed Storage Management

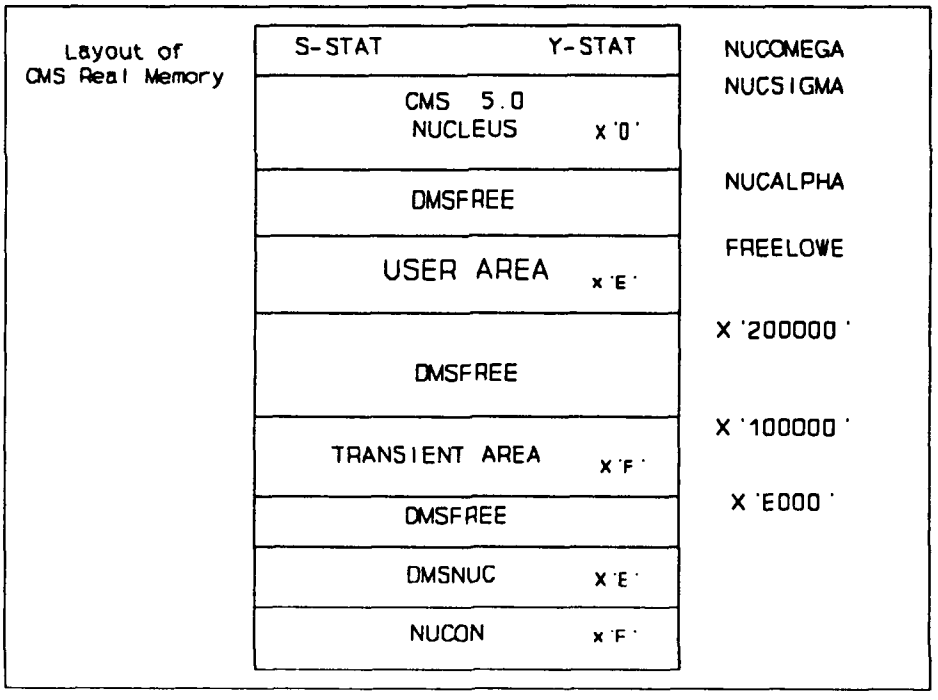


FIGURE 2.4.1

It would be wasteful of Free Storage Area to allocate a pagetable entry to every page defined for a particular VM. Instead, CP only sets up enough page table entries, and allocates enough real storage, to point to the minimum storage needed by a machine when it is IPL'd. As more space is needed, SVC 203 is used to request additional storage. This is similar to the standard IBM 370 GETMAIN and FREMAIN SVC calls. Referring to Figure 2.4.1, the S-STAT, Y-STAT and CMS NUCLEUS areas are brought into memory the first time someone IPLs CMS, so their page table entry pointers are to the same entries for each VM running CMS. NUCON contains specific storage area for this VM, such as registers and CMS data areas. The Global Control Block also resides in NUCON. S-STAT and Y-STAT contain file system directories for shared devices, and are shared with all other VMs. The user area contains user generated data and storage space for programs loaded into the transient area using the LOAD command. The transient area contains programs loaded using the LOAD command.

The three DMSFREE areas are used for allocating additional storage. The lowest DMSFREE area is used for nucleus extension programs. The middle DMSFREE area (called high) is used to allocate storage for user programs that ask for additional memory. The highest DMSFREE area, which doesn't have any special name, is used by the system.

Input/Output to Attached Devices

While a user program can make use of any IBM 370 I/O routines, including writing its own Channel Command Programs, most CMS users rely on the virtual I/O facilities that CMS provides. Each user has PUN, PRT and RDR devices allocated at IPL, and CP handles these as files on its own SYSRES disk. Each user can attach to his own minidisks and to those of other users who allow him to attach to their minidisks. At IPL, CMS creates an Active Disk Table for the user's own minidisks described in the CP Directory. As other minidisks are ACCESSed, new table entries are created. CMS uses special DIAG commands that direct CP to perform minidisk I/O, using virtual CCW programs built by CMS. The spool files have different DIAG commands to CP that perform synchronous I/O using SIO and TIO commands. All I/O is virtual, using virtual channel, controller and device descriptor blocks created by CP as devices are accessed.

RACF

The Resource Access Control Facility (RACF) is used to verify user identities, grant access to resources, record auditable events, and generate audit and administrative reports. RACF executes in the RACFVM service virtual machine. Typically, RACFVM executes as a disconnected VM; there is no user logged onto this VM. A RACF backup service machine, called RACMAINT, is also maintained to perform RACFVM's operations during administrative maintenance or RACF recovery. RACMAINT can also used to generate administrative reports.

RACFVM is also responsible for autologging RACFSMF when RACF's audit log becomes full. RACFSMF archives the audit log and empties the minidisk upon which it resided. A complete description of RACFSMF's operations is given on page 81, "Audit Mechanisms". Also, see page 62, "RACF Interface".

The privileges that a user has within the system and the ability to access RACF protected resources is based upon three gating factors: the user's RACF attributes, the user's group authority, and the user's resource access authority with respect to a protected resource. All of this information is stored within RACF profiles.

RACF maintains a data base in which it stores profiles. Profiles are used by RACF to control access to resources (see page 74, "DAC"). Specifically, profiles contain attributes and access authorities for every user and resource. A description of RACF characteristics including profiles, RACF classes, the data base, resource access authorities, group authorities, and attributes follows. The RACF program components are then described.

RACF Profiles

RACF profiles contain the description of attributes and access authorities used to restrict access to all RACF protected resources. There are four types of RACF profiles: the user profile, the group profile, the connect profile, and the resource profile. A description of each is given below:

USER Profile

A user profile defines an individual user to RACF. A user profile is created when the RACF command ADDUSER (define a user to RACF) is issued and altered when the ALTUSER command is issued. The user profile contains the userid, RACF password, user level attributes, the name of the user's default group, events to be audited, the password change interval, and the name of the profile used to create new profiles for the user.

GROUP Profile

A group profile defines a group; a group is a collection of related users. A group profile is created when the RACF command ADDGROUP is issued and altered when the ALTGROUP command is issued. A group profile contains a list of the group's sub-groups, members, and a list of group-authorities for each member. The profile also contains the name of the profile used to protect newly defined group resources.

CONNECT Profile

When a user is defined to RACF and when the user is added to an additional group with the CONNECT command, a connect profile is created. This profile contains the group-level attributes that are valid within the group for the user.

Resource Profile

A resource profile is created for each RACF protected resource. It contains the resource name, a list of userids including the Universal Access Authority (UACC) (see page 76, "Universal Access Authority"), their access authority, and certain statistics of audited events (e.g. the number of times that the profile was referenced by RACF).

There are two types of resource profiles. The first, a discrete profile, protects a single resource with unique access or auditing requirements. The other, a generic profile is intended to protect several resources with similar naming constructs and similar access

requirements. With generic profiles, an individual resource does not have to be specifically named to RACF to assure that it is protected. It is only required the name of the resource match one of the generic profiles. Additionally, a generic profile will remain in the data base even after the last resource it protects is deallocated.

For administrative purposes, resource profiles are referred to as data set profiles or general resource profiles. Each type of resource profile can be discrete or generic. Data set profiles protect MVS data sets (a file). General resource profiles protect all other resources (e.g. minidisks, terminals, tape volumes). This distinction is made because data sets are created dynamically and thus profiles to protect them are also created dynamically. On the other hand, the number of general resources remain relatively fixed as does the number of general resource profiles. For VM/SP with RACF, data set profiles are not used; only general resource profiles are used.

Additionally, every profile is owned by a user or group. By default, the owner is the user or group who created the profile, or the user or group specified in the OWNER parameter of the RACF ADDUSER, ALTUSER, ADDGROUP, or ALTGROUP command. A complete description of users and groups can be found on page 78, "Ident. and Auth.".

RACF Classes

A RACF class is a collection of resources or users with similar characteristics. All protected resources and users must be assigned to a class when its profile is defined.

When RACF is installed, IBM supports forty three RACF resource classes. Of these forty three, the following apply to VM/SP with RACF: FIELD, TERMINAL, GTERMINL, VMMDISK, VMNODE, VMRDR, VMBATCH, VMCMD, GLOBAL, GMBR, SECDATA, TAPEVOL, SCDMBR, VMBR and VMEVENT. If these classes do not meet the needs of an installation, specific classes can be defined. In either case all valid resource class names, except the DATASET class name for MVS, are stored in the Class Descriptor Table (CDT). RACF also supports the USER and GROUP class to which users are assigned. However, these two classes are not stored in the CDT since they do not refer to resources.

The CDT is referenced every time a user attempts to access a resource. If the class is undefined, access is denied. Class names can also be added to the Global Access Table (GAT) so that a resource from the class can be made a public object. For a complete description of access authorization and the GAT, see page 74, "DAC". For VM/SP with RACF, this CDT check cannot be deactivated. However, on MVS a user with the SPECIAL attribute could deactivate it.

RACF Data Base

The RACF data base contains all RACF profiles and thus all access control information. The RACF data base is a single logical data base owned by (i.e. CP Directory ownership) RACFVM. The data base can physically reside on a single minidisk or multiple minidisks in which case it is referred to as a multiple RACF data base.

The data base holds definitions of users, resource authorizations, and RACF attributes. RACF uses the data base when a user is defined to RACF, when the user subsequently enters the system, when access to a RACF protected resource is defined or altered, and when a user accesses a protected resource. Specifically, I/O requests from RACF to the data base is controlled by the RACF manager (see page 64, "RACF Mngr").

Resource Access Authority

To access a RACF protected resource, a user or group must be listed in the resource profile (either generic or discrete) and have a specific access authority. The specific forms of access for all users, groups, and the UACC are:

ALTER
For discrete profiles, users or groups with ALTER authority have full access and control over the profile and the resource. For generic profiles, full access to resources protected by the profile is granted, but no control on the profile is exerted. Only the user who owns the profile, a SPECIAL user, or a user with group-SPECIAL within the scope of the profile's group, can modify the profile.

CONTROL
CONTROL is used only on MVS systems but is listed here for completeness.

UPDATE
Users or groups assigned UPDATE can observe and modify the contents of the object.

READ
Users or groups assigned READ can only observe the contents of the object.

NONE
Users or groups assigned NONE cannot access the object.

Group Authorities

For a complete description of RACF groups, see page 78, "RACF Groups".

A group authority is given to each member of the group based upon the user's responsibility within the group. Group authorities are contained in the group profile defined for each group. Once assigned, group authorities allow the users of a group to use, manage, and assign resources to the group. Group authorities can only be assigned by a user with the (user level) SPECIAL attribute, or group-SPECIAL attribute but then only within the scope of the group. The various group authorities are:

USE
A user with this authority is allowed to enter the system under control of the group, to access protected resources to which the group is authorized, and create RACF defined objects.

CREATE
The CREATE authority contains the capabilities of USE and additionally the ability to create RACF group protected resources.

CONNECT
The CONNECT authority contains the capabilities of CREATE. A CONNECT user can also assign a previously defined RACF user to the group and distribute USE, CREATE, or CONNECT authority to the newly assigned member of the group.

JOIN
JOIN contains the privileges of CONNECT. A JOIN user can define new users to RACF, assign any group authority, and define new groups to RACF. A new user can only be defined if the JOIN user has the CLAUTH attribute for the USER class and newly defined groups will be a sub-group of the group to which the JOIN user is assigned.

RACF Attributes

Attributes are additional privileges which can be assigned to each RACF user. Attributes are distinguished as either User-level attributes or group-level attributes. User-level attributes are stored in user profiles and the specified privilege applies globally; it is independent of the user's current group. Group-level attributes are stored in connect profiles and only apply to the scope of the group. See page 79, "Group Scope", for a complete description of groups and the scope of a group.

User Level Attributes

SPECIAL
Users assigned the SPECIAL attribute are able to issue all RACF commands, except those reserved for the user with the AUDITOR attribute, and control all profiles in the RACF data base.

AUDITOR
The AUDITOR attribute will allow a user to specify RACF auditing options, list auditing information, list all profile information available to the SPECIAL user, control the SMF data set, and execute the Data Security Monitor (DSMON) and RACF report writer. (see page 64, "RACF Audit").
This attribute can only be distributed by a user with the SPECIAL attribute.

OPERATIONS
The OPERATIONS attribute gives a user full authorization to all RACF protected resources unless specifically excluded (i.e. given NONE access authority) or limited by the resource profile protecting the resource. Additionally, to gain access, the CDT entry to the class which the resource is assigned must allow OPERATIONS access (see page 76, "OPERATIONS Attribute Checking").

The OPERATIONS attribute can only be distributed by a user with the SPECIAL attribute.

CLAUTH

The CLAUTH or class authority attribute is assigned on a class by class basis. That is, if a user has the CLAUTH attribute to a RACF class, then that user is able to define RACF profiles in that class. The CLAUTH attribute can be assigned for all classes in the CDT and the USER class. To define new RACF users, in addition to being assigned CLAUTH to the USER class, the CLAUTH user must also be the owner of a group or have JOIN authority.
The CLAUTH attribute can only be distributed by a user with the SPECIAL attribute or by a CLAUTH user but only for the classes the user is authorized to.

GRPACC

When a user with GRPACC (group access) attribute defines a group profile, users within the group are given UPDATE authority to the resource protected by the newly defined profile. This group authority is only for MVS.
The GRPACC attribute can only be distributed by a user with the SPECIAL attribute or the owner of the user's profile.

ADSP

The ADSP attribute will automatically define to RACF a discrete profile that will protect a permanent data set. This group authority is only for MVS.

The ADSP attribute can only be distributed by a user with the SPECIAL attribute or the owner of the user's profile.

REVOKE

The REVOKE attribute is used to prevent a user from accessing the system. This attribute can only be assigned by the owner of this user or by a user with the SPECIAL attribute.

Group Level Attributes

The SPECIAL, AUDITOR, OPERATIONS, GRPACC, and REVOKE attributes can all be assigned to the group level. The function of the attribute is the same as described above, but the capability is limited to the assigned group and its sub-groups. At the group level, the attributes are referred to as group-SPECIAL, group-AUDITOR, group-OPERATIONS, group-GRPACC, and group-REVOKE.

RACF Program Components

RACF is divided into six major components that perform all of RACF's operations. They are: RACF initialization, the RACF interface, the RACF command processors, the RACF manager, RACF utilities, and RACF auditing. A discussion of each component is given below:

RACF Initialization

There are several EXECS which are used during RACF installation and also for the installation of the data base, either a new or updated data base. These EXECs are described in the *RACF Program Directory for VM Installations* [12].

RACF Interface

CP and other virtual machines can request RACF operations or alter data contained within RACF (i.e. the data base and certain tables). All RACF requests are channeled through the IUCV support code (see page 41, "Inter-User Comm. Vehicle") initiated by the primary RACF machine, usually RACFVM. A request can be sent by either CP or a virtual machine. Once received, the request is passed to CMS Sub Tasking (CST).

CST is an enhanced version of CMS which executes in RACFVM and RACMAINT. CST is delivered as part of RACF and provides an interface to RACF for CP and virtual machines. CST interprets the VM/SP specific requests so that the proper RACF macros can be issued and the RACF request can be processed. Once completed, CST relays the results back to CP or the virtual machine through the IUVC support code.

CST is responsible for accepting, calling, and responding only to VM/SP specific requests. This means that requests applicable to MVS cannot be called by CP or a virtual machine, although the code does reside within RACFVM and RACMAINT.

RACF also maintains a role in checking access to, and in auditing the use of, the security relevant CP commands, DIAGNOSE instructions, spool activities, and communication between VMs. The specific events that CP can process for a VM are listed in the Access Control Interface (ACI) bit map. The ACI bit map is CP's gateway to RACF. CP refers to this bit map every time one of the listed events occurs. If the "control" entry is set for the event, CP calls RACF through CST to validate the request. If the "audit" entry is set, CP again calls RACF.

There are two types of ACI bit maps, the individual ACI bit map and the system ACI bit map. A CP Directory entry is set when a VM has an individual ACI bit map. There can be many individual bit maps, one assigned to each individual VM. When a VM that has an individual ACI bitmap is initialized, CP places the current copy of it into the Free Storage Area, and adds a pointer to its location in the VMBLOK. If an individual bit map exists for a VM, CP refers to it and not the system bit map. The individual bit map can also be used to exempt a VM from RACF access and auditing control. For a C2 system, all untrusted subjects must be under the control of RACF and cannot be exempt from access control.

Essentially, the individual bit map provides a mechanism by which unique access and auditing control can be placed upon each individual VM. See page 81, "Audit Mechanisms" for a further discussion on how the ACI bit map applies to auditing on VM/SP with RACF.

VMs cannot access the system or their individual ACI bit map directly. Instead, their requests are routed by CST to the RACF command processor (see the next section below). Additionally, CST only allows CP to execute privileged RACF routines, namely SVC macros. These SVCs execute with privilege local to the RACF machine, RACFVM or RACMAINT. For example, only CP can call the RACINIT SVC to provide user verification. This macro requires privilege so that it can reference authentication data contained within the RACF data base.

An IUCV connection to RACF can only be established by the primary RACF machine, typically RACFVM. This means that a virtual machine cannot initiate an IUCV session with RACF, but can only request that RACF establish one to it. This is done when the VM issues the RACF command to a RACF, establishing a "RACF command session". Similarly, RACFVM establishes a connection to CP once autologged by AUTOLOG1.

The RACF connection to a VM will remain until it is terminated by the VM, whereas the connection to CP is permanent. However, there may be an uncommon situation where the IUCV connection to CP is lost (e.g. a RACF program check occurs). In this situation, RACF is down and the interface severed. CP will then toggle the ACI invalid bit on. As stated in the *C2 Security Guide* [28], the SEVER=Yes option must be set for the C2 configuration. This option will cause the system to abend when RACF is severed and not violate the security policy of the system. Additionally, this option allows RACFSMF to break the IUCV connection between the primary RACF virtual machine and CP when the audit log becomes full and it can no longer be archived. As discussed above, CP will note the loss of RACF, set the invalid bit, and abend.

In VM/SP, there are three ways to recover the RACF interface. An operator could log onto AUTOLOG1 and re-execute the RACF autologging procedure. An operator could log onto RACMAINT, LINK to the minidisks containing the RACF data base, and FORCE an IUCV connection to CP. This will cause RACMAINT to become the primary RACF machine. Lastly, the operator could log onto RACFVM, fix the problem, and re-establish the IUCV connection with CP via the RACSTART EXEC.

RACF Command Processor

All VMs can issue RACF commands, although not all commands may be issued by all VMs. RACF commands can affect or query data within RACF. To issue commands, a RACF command session must be established from CMS as described in the "RACF Interface" section above. Once established, RACF will determine if the request is valid, based upon syntax and the VM's RACF privileges, before the request is completed.

RACF Manager

The RACF manager performs operations on the RACF data base at the request of the RACF commands, RACF utility programs, and RACF SVC macros. Specifically, the RACF manager processes nine requests: add a profile, alter a profile, alter a profile in place (size of the profile does not change), delete profile, delete a member of a tape volume set from a TAPEVOL resource class profile, locate a profile, find next profile of a given type, find next profile in collating sequence, and rename a profile.

RACF Utilities

The RACF utilities are used to maintain, modify, and monitor the contents of the RACF data base. The utilities are called by their corresponding EXEC from RACFVM, RACMAINT, or in certain cases from a VM which has identical links as RACMAINT and has the proper RACF attributes. An example of a utility would be the RACUT100 EXEC which calls ICHUT100. ICHUT100 lists all occurrences of a userid or group name associated with a resource profile. It can be invoked by a VM possessing the SPECIAL, group-SPECIAL, AUDITOR, or group-AUDITOR attributes, or by requesting the occurrences of onself. The EXECs for all utilities are listed in the [34]

RACF Audit

RACF has the ability to audit events when a user interacts with resources. Auditing takes place when users or groups interact with resources in a manner which meets the selected auditing options indicated in the ACI bit map. Additionally, certain statistics can be recorded in resource profiles to determine the profile's usage. When RACF cuts an audit log entry, a System Management Facilities (SMF) record is written to a CMS file. The audit reduction tool, RACF Report Writer, can be used to print a readable audit log. For more information about RACF auditing, see page 81, "Audit Mechanisms".

Virtual Machine/Directory Maintenance Licensed Program

The Virtual Machine/Directory Maintenance Licensed Program (DIRMAINT) allows users to make changes to their entries in the CP Directory. DIRMAINT administrators, called staff members, also use DIRMAINT to control system resources such as the configuration of virtual machines and the allocation of minidisks extents. DIRMAINT warns system administrators if they attempt to overlap the physical extents of minidisks belonging to different virtual machines. A description of VM directories, the DMPP virtual machines, and DMPP operations follows.

Virtual Machine Directory

The CP Directory contains an entry for each authorized user. Specifically, the directory contains the USERID, the DIRMAINT password which is the same as the RACF password, accounting information, class priority, the CP privilege class, and a description of the user's VM configuration. The user's VM configuration indicates to CP the maximum virtual storage,

the spool addresses, the location of the VM's minidisks, their size, and which minidisks to auto link to at IPL.

The VM directory is represented in two forms, the directory source form and CP Directory form. These represent the source and object format of the information. The source form is contained on DIRMAINT 196 disk as a system file. This source file is subsequently broken into cluster files that contain an average of 100 user definitions. These definitions are the actual control statements that compose each user.

The CP Directory is a processed version of the source directory. Processing by the DIRMAINT DIRECT service program translates the source form into control blocks which are used by CP during the creation of the user's VM (i.e. at logon). CP validates the user identity and builds the proper VM based on the configuration and operational characteristics in the CP Directory.

The DMPP Virtual Machines

DMPP maintenance functions are performed by two virtual machines which execute as two permanently disconnected virtual machines; users do not log onto them. These machines, DIRMAINT and DATAMOVR are autologged at IPL time and operate (idle) in a disconnected state until a DIRMAINT operation is requested from a user's VM to DIRMAINT via VMCF. Such requests come from VMs in CP READ state or from CMS when the user either enters a DIRMAINT session by typing DIRM, or when the user types "DIRM command" to execute a single DIRMAINT command. For example, "DIRM DIRECT" can be issued from a CP session to recompile the CP Directory without entering a DIRMAINT session. This user interface is similar to the interface for RACF. If the DIRMAINT VM fails (abends) processing continues as normal. CP is not effected. The only result of this failure is that changes to the directory cannot be performed until DIRMAINT is restarted.

The specific functions of DIRMAINT and DATAMOVR are discussed below.

DIRMAINT Virtual Machine

DIRMAINT owns and manages the source directory file and has the authority to rebuild and backup the working version of the CP Directory. DIRMAINT is responsible for processing all directory requests, maintaining the DIRMAINT audit log, and assuring that only the authorized issue privileged DMPP commands. It can also generate a listing of free extents of DASD volumes used to contain minidisks.

DIRMAINT commands originate in the user's VM and, after pre-processing, commands are transmitted to the DIRMAINT VM via the CP SMSG command; CP SMSG uses VMCF. If the DIRMAINT VM is busy, the command remains in the command queue until it can be executed. Bulk data can also be transferred using a spool file. Once the command is parsed, the DIRMAINT VM replies to the user via CP MSGNOH command or the CP SMSG command.

DATAMOVR Virtual Machine

The DATAMOVR VM processes DIRMAINT requests to copy or format CMS minidisks. Operands destined for the DATAMOVR VM are sent to the DIRMAINT VM first. After processing, DATAMOVR will route its responses to the DIRMAINT VM so that transmission to the originating VM can be made. DATAMOVR is a support machine for DIRMAINT.

DMPP Operations

Users are able to issue a limited number of commands to query and change information stored in their CP Directory entry. For example, a user can query the last time he changed his password, or change the initial system that is IPLed when he logs on. However, changes that affect system resources are modified only by authorized DIRMAINT staff members who possess a DMPP privilege. Staff members can perform three classes of operations: directory administration operations, control and monitoring operations, and maintenance operations.

The authority to issue one of these privileged operations is determined by the user ID designation in the DIRMAINT DATA file. Note, that this file also contains the required statement, DISK_CLEANUP=YES to assure that all released minidisks extents are overwritten (see page 77, "Object Reuse").

A privileged user can be designated as follows: DIRMAINT member, DIRMAINT substaff member, DIRMAINT owner, DIRMAINT operator, or DIRMAINT password monitor. Each privilege gives the user the authority to issue a specific subset of the DMPP commands. The capabilities of these privileged users are as follows:

DIRMAINT Staff Member
This user is authorized to issue all DMPP privileged operations except the RLDCode operation (see "DIRMAINT Owner" below).

DIRMAINT Staff Substaff Member
This user is able to issue a subset of the privileged operations and is restricted from operations that result in directory changes, or affect system control and resources.

DIRMAINT Owner
This user is the only user who can issue the RLDCode operation to update the DIRMAINT executable code. The DIRMAINT Owner is responsible for installing and maintaining DMPP.

Password Monitor
This user has the privilege to issue the DIRMAINT commands to audit, generate, set, and change CP passwords of users. Since only the password in the RACF profile is used for authentication, this operation does not perform a useful service.

System Operator
This DIRMAINT user can issue the privileged instructions to terminate the DIRMAINT VMs, and back up and terminate the automatic back up of the DIRMAINT DATA file to tape.

DIRMAINT Logs

Virtual Machine/Directory Maintenance Licensed Program maintains several files to log events. In particular, DIRMAINT uses a spooled console log to maintain a running account of all incoming transactions and the command DIRMAINT issues to process them. This creates an audit log of all security relevant events that are connected with DIRMAINT. This is the only audit log in the VM/SP with RACF system that is not maintained by RACF.

VM-TAPE

VMTAPE 4.1 is intended to provide the Security Policy requirements for tape volumes. These requirements are DAC and object reuse. For a complete discussion on DAC for tape volumes, see page 74, "DAC". Tape volume object reuse is discussed on page 77, "Object Reuse". This section explains how VMTAPE works.

To understand how VMTAPE works one must understand how tapes are accessed in VM. Without VMTAPE, a tape is accessed by performing the MOUNT and ATTACH commands. The MOUNT command sends a message to the tape operator requesting a specific volume/serial number (volser) or a scratch tape to be mounted on a specific tape drive. The tape operator must physically mount the tape on a tape drive and then the user ATTACHes to the drive.

VMTAPE acts as the security monitor between the untrusted VM and the tape operator. It uses a Tape Manager Catalog (TMC) to keep the latest statistics on each tape, provides auditing and calls RACF on each tape access to check for DAC. The tape operator performs exactly the same steps for a VMTAPE MOUNT as for a normal MOUNT. VMTAPE performs the ATTACH command on behalf of the user. VMTAPE performs the following operations:

- A VM user must establishes a VMCF communication link to the VMTAPE service machine and transmits the request for a VMTAPE MOUNT.

- If the MOUNT command is for a scratch tape, VMTAPE either selects a volser using the automatic scratch selection facility or sends a message to the tape operator to type in a volser of a scratch file.

- If the MOUNT command requests a specific volser, VMTAPE verifies the user's authority by calling RACF. If the user is not authorized to access the volser,VMTAPE rejects the MOUNT command and sends a message to the user via VMCF.

- If the mount request is not rejected, VMTAPE sends a message to the tape operator to mount the specific volser requested on a tape drive that VMTAPE has chosen.

- The tape operator mounts the tape on the specified drive.

- VMTAPE verifies[1] that the volser mounted is the one expected and determines whether there is a write ring in place, depending on the user's request. Then VMTAPE updates the TMC and the Audit file, issues an ATTACH command to the tape drive on behalf the requesting VM user and sends messages to the user and the operator indicating that the mount is complete.

- The VM user uses the tape.

- The VM user controls that tape and drive until he invokes a CP DETACH command. Only then will VMTAPE reassign the drive.

The following observations are pertinent to the security of tape volumes on this system:

- Once VMTAPE has relinquished control, the user(with read/write access) of the tape can modify all data on the tape (including the label).

- VMTAPE maintains the tape label information in two places, on the tape and in the TMC. Therefore the label information on the tape does not have to be assured because the tape's label is maintained in the TMC. Destruction of the tape label can only cause the VMTAPE system to refuse any subsequent mounts.

- If VMTAPE fails the TCB would no longer be able to support access to tape until VMTAPE is logged back on. There is no known failure of the TCB when VMTAPE fails. Furthermore, because of the design of VM/SP, the failure of any component is likely to be effectively handled. This is because work is accomplished through message passing rather than invocation. If a process is unavailable, CP enqueues the messages or tells the sending process the communication path has been severed. This provides a more robust interface.

ISPF

Interactive System Productivity Facility, ISPF can be used to perform security and group administration tasks. ISPF allows specific RACF commands and their options to be entered interactively from ISPF panels (menus). All RACF commands except RVARY and RACFRW (report writer) can be issued from ISPF panels. Only the commands to add, change or delete

---

[1] If the tape is standard labeled this information is verified using this label information. If the tape has no label then the tape operator is requested to type the physical label twice to identify the tape.

a VM user or minidisk require dual registration, so only these commands must be invoked using the ISPF panels. When a RACF command requires a corresponding DIRMAINT command, ISPF issues that comand as well, assuring that all users are defined to both RACF and the CP directory at the same time.

The Dual Registration Facility that is provided is a set of REXX EXECs that drive a series of ISPF data entry panels. The REXX EXEC that controls the display of the data entry panel runs in the VM of a System Administrator with the appropriate RACF and CP privileges. The REXX EXECs reside on a minidisk belonging to RACFVM and can be invoked by a RACF privileged user who is running a RACF session in his own VM. The REXX EXEC will issue the appropriate RACF command first (in realtime) and if the return code from RACF is good it will then issue the corresponding DIRMAINT command. DIRMAINT may queue the request, depending on the activity of the DIRMAINT service machine. After each DIRMAINT command that alters the source directory a DIRM DIRECT command is issued to place the changes online. See page 66, "DMPP" for a discussion of DIRM DIRECT.

DIRMAINT does not issue a return code for a command issued in the REXX EXEC, but DIRMAINT will issue a message which will display on the terminal (console) of the administrator's VM. It is possible that the RACF command worked but the DIRMAINT command failed; in that case the administrator would have to undo the RACF command (using the ISPF panels, of course) or find out why the DIRMAINT command failed and correct the problem and re-issue the DIRMAINT command.

In no event will there by a security problem because RACF performs its access checks by checking for the correct profile in the RACF database and also checking for appropriate entries in the CP Directory. Access is never granted if the CP Directory entry is incorrect or missing. For example, if the ISPF menu facility is used to add a new user, the RACF ADDUSER command is first issued by the REXX EXEC. If the user is successfully added to the RACF database, then the DIRM ADDUSER command is queued to the DIRMAINT SVM. The DIRM DIRECT command is then issued, which will rewrite the CP Directory from its source form, whether or not the changes were made, since the REXX EXEC has no way of knowing whether the command to DIRMAINT was successful. In the event that RACF added a new user, but DIRMAINT failed to add the new user, the user cannot log on or, even if he could, he could not access any objects. This is clear from the discussion of access checking on page 75, "Ownership Checking", where CP first checks the CP Directory for the VM to see if it is the owner of an object. If no CP Directory entry exists, then RACF is not even called and no further checks are made. Similar arguments can be made for all other instances where RACF has an entry in its database but no corresponding entry in the CP Directory.

If the ISPVM Service VM went offline, then attempts to run the EXECs stored on the RACFVM's minidisks would fail. Since all operations that require dual registration must be run from ISPF panels, none of these operations could be performed until the ISPVM was logged back on.

TCB Protected Resources

VM/SP with RACF enforces separation of processes acting on behalf of users from a variety of named objects in the system. In addition, the TCB ensures that storage objects are never readable by a new user when they are reallocated. The following sections describe the subjects, named objects and storage objects in the system.

Subjects

There is only one type of subject in the VM/SP with RACF system: the Virtual Machine. A virtual machine can best be defined as any entity represented by a VMBLOK in the Free Storage Area, except for the VMBLOK that represents CP itself.

In addition to user Virtual Machines, there is a set of VMs that are termed trusted subjects because, even though they run in the same space as user VMs, they are allowed access to security relevant system data. (See page 28, "TCB Component List".) They also run with an appropriate CP privilege class. Some of these VMs are logged on by AUTOLOG1, which is the first VM that is entered on the system when it is IPLed, others by AUTOLOG2, which is shipped with RACF and prevents VMs that may perform security relevant events from logging on until RACF is running. The others are logged on as the various parts of the TCB are IPLed during system startup. The correct procedure for starting these VMs is referenced in the *Security Guide* [28]. (See page 73, "Privileges".) The list of these trusted VMs is in the section on software TCB components. (See page 28, "TCB Component List".)

Every other VM represented by a VMBLOK in the FSA is acting on behalf of a system user who has logged into the system and been verified by CP and authenticated by RACF. These are the subjects whose access to named objects is mediated by RACF.

Objects

VM/SP with RACF provides users with direct or indirect access to eight named objects upon which DAC is placed, and five storage objects which must meet the object reuse requirement. This section provides a definition of these objects and then lists them as named objects, storage objects, or both. In addition, two public objects[1] are identified.

Minidisks
: Minidisks are a logical representation of a DASD device which is linked to a virtual machine; interactions with this virtual device are converted to real DASD storage interactions by CP. This real storage is usually a partition of real DASD device and

---

[1] Definition: A Public Object is an object for which the ADP system implementation permits all untrusted subjects to perform non-modifying operations (e.g. read, execute) on the object's contents and attributes, but permits only trusted subjects to modify the object's contents or attributes, and permits only trusted subjects to create or destroy such objects.

is called a physical extent. The boundaries of these extents are listed in the CP Directory so that the logical assignment can be made.

Spool File
The spool file is a system controlled DASD area used by virtual machines to send and receive data. This area is controlled so that virtual machines can only send data to authorized virtual machines and only receive data sent to them. This access control occurs when a virtual machine uses the only mechanisms available to access the spool file: the virtual reader, the virtual punch, and virtual print.

Tape Volumes
A tape volume represents a collection of data stored on a magnetic tape. Users may be authorized to control the contents and the access to tape volumes. See page 29, "VMTAPE".

VMCF Buffers
On page 40, "VMCF", there is a complete description of the buffers involved with this type of IPC.

IUCV Buffers
On page 41, "IUCV", there is a complete description of the buffers involved with this type of IPC.

VCTCA Channels
On page 44, "VCTCA", there is a complete description of the virtual channels involved with this type of IPC.

APPC/VM Buffers
On page 44, "APPC", there is a complete description of the buffers involved with this type of IPC.

MSG Buffers
The MSG and SMSG commands can be used to send IPC messages to another logged on user or to a server VM, respectively. CP formats the message as either a VMCF or IUCV message and transmits it to the recipient, provided the CP directory entries for the sender and recipient allow such a message. See the discussion of VMCF and IUCV IPC referred to above.

Temporary Minidisks
Temporary Minidisks (T-disks) are similar to minidisks. Both are logically assigned to a virtual machine and mapped to physical extents by CP. However, T-disks allocated to a requesting VM during its existence are returned to the system pool at its termination, or when relinquished by the VM. The system pool is part of CP's unused physical DASD area

Virtual Storage

Virtual Storage is the conceptual representation of memory for each VM in which tasks related to the VM logically reside. It is mapped by DAT to real storage so that these tasks can executed.

Discontiguous Saved Segments

An entry in the CP Directory defines an area of real storage as a DCSS. This area, which may include data or executable code, is incorporated into the address space of a VM at a location higher than, and discontiguous with, the VM's previous maximum address. This area is accessible by all virtual machines through the system's address translation mechanism. Specifically, the common segment bit of DAT allows all virtual machines to read and execute the DCSS. However if the virtual machine attempts to modify any page in its virtual storage that maps to the DCSS, a fault will occur which creates a new segment table for that virtual machine. Thus, the modified area will be mapped by the new segment table. A description of Discontiguous Saved Segments (DCSS) is given on page 32, "CP Memory Management".

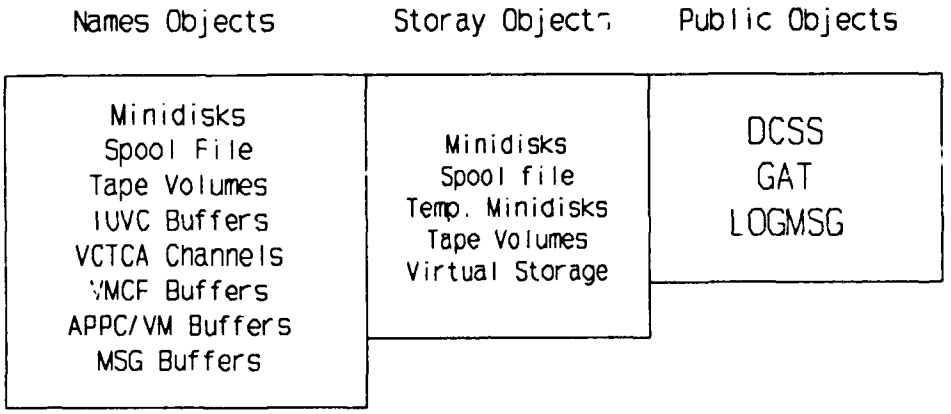
Minidisks in the RACF Global Access Checking Table

A complete description of the RACF Global Access Checking Table (GAT) is given on page 75, "GAC".

Log Messages (LOGMSG)

The system Log Message may be set by the Operator or the System Administrator using the SET LOGMSG command, a class B command. The LOGMSG is kept in the WARM START area of the system disk SYSRES, which means that a COLD START will clear it. It can also be cleared or changed by Operator action. The current LOGMSG is displayed on each user's terminal at every logon. A QUERY LOGMSG command will also display the LOGMSG on the user's terminal. The existing IPC services used for MSG are used to transmit the LOGMSG to the user's terminal, but it is not possible, or desirable, to put the same kind of discretionary access control on the LOGMSG as on MSG buffers, so they are considered to be public objects.

The following chart designates each object as named, storage, or public, as appropriate:



OBJECT CHART

TCB Protection Mechanisms

Privileges

CP Privilege Classes

VM/SP with RACF supports privilege classes in order to enforce the principle of least privilege. These classes are used to restrict certain CP commands to certain privilege classes. There are initially seven privilege classes distributed with the system, designated A through G, such that each class is restricted to the use of certain CP commands. However, a user may be given more than one privilege class and is allowed to use any command allowed by any class associated with his logon ID. Class G users are general users with no privilege; the other predefined classes generally correspond to a privileged user, such as Class A which corresponds to the operator function.

The system administrator can define additional classes for accounting purposes, or to further restrict which CP commands users assigned to these classes can use.

RACF Privileges

RACF attributes and group authorities are the two types of privileges available from RACF. There are various attributes and group authorities that offer a different degree of capability to allow subjects to alter access to resources, select auditable events, and administer group control. For a complete discussion of the various attributes, see page 60, "RACF Attributes",

and for a complete discussion of the RACF group authorities, see page 60, "Group Authorities".

DIRMAINT Privileges

Staff members are DIRMAINT users with certain privileged capabilities that allow them to modify and monitor the CP-Directory. Staff members are designated as follows: DIRMAINT member, DIRMAINT substaff member, DIRMAINT owner, DIRMAINT operator, or DIRMAINT password monitor. A complete description of DMPP privileges is found on page 66, "DMPP Operations".

ISPF Privileges

ISPF has no special privileges related to it. However, certain privileges must be possessed by a user so that ISPF can execute the requested operation. For example, the user must possess the proper RACF privileges when ISPF issues a requested RACF operation.

VMTAPE Privileges

VMTAPE has two special privileges which the VMTAPE administrator can authorize for users. They are Bypass Label Processing (BLP) and No Label (NL) privileges. BLP allows the user to ignore the tape label. This is used when a foreign tape has a tape label which is unreadable to the system. NL privilege allows the user to create or process a tape which does not have a label. These capabilities are considered privileged because only Standard Label (SL) tape processing has been specified by the vendor as meeting C2 requirements.

These various types of privilege, and their interactions, are discussed in the Security Guide [28] .

Discretionary Access Control

Discretionary Access Control in enforced by interactions between CP and RACF, and by interaction between RACF and other Service VMs. CP is able to separate each subject by mapping it to a virtual machine. This mapping is controlled by Dynamic Address Translation (DAT) and its use assures that subjects are unable to directly gain access to storage belonging to another VM.

Whenever a virtual machine requests access to an object (minidisk, spool file, or tape volume), CP first checks the CP Directory to determine if the virtual machine owns the requested object. If not, CP calls RACF to determine if access can be granted. RACF responds based upon the subject's attributes, group authority, and access authority contained in the object's profile. CP then makes the final decision.

The procedure for access checking is described fully below. This procedure is performed for minidisks, spool files, and tape volumes.

Ownership Checking

When a subject attempts to gain access to a minidisk or spool file (i.e. another VM's reader), CP checks the virtual machine's entry in the CP Directory to determine if the subject is attempting to access an object it owns. If CP determines that the subject has ownership, then access is granted. If the subject does not own the object, CP references the appropriate ACI bit map, either individual or system. The "control" entry for the LINK and SPOOL (along with STCP, TAG, and TRANSFER) commands are set by default and therefore, RACF will always be called when access to a non CP Directory owned minidisk or spool file is attempted. In the case of attempt to access a tape, the VMTAPE VM calls RACF directly, rather than going through CP. However, the same access checks are performed as described below.

Class Descriptor Table Check

RACF first determines if the object has been assigned to a RACF class contained within the CDT. If it has, RACF processing continues. Otherwise, RACF fails the request.

Global Access Checking

After checking the CDT, RACF determines if the object has been assigned to the global access table (GAT). If assigned and the attempted access, either READ, UPDATE, CONTROL, or ALTER, does not exceed the listed authority, RACF will permit access, regardless of the subject's identity. Access can only be granted by the global access check; it can not be denied. If the attempt fails, RACF continues access checking. Additionally, RACF does not write any auditing records or maintain profile statistics when global access checking permits access to an object.

Profile Search If the GAT check failed, RACF then searches the RACF data base for the resource profile that protects the object. The data base is internally structured so that the most specific profile name will be found first. Therefore, when RACF searches the data base, a discrete profile will be found before any generic profiles. If no profile is found, RACF fails the request.

Security Classification Checking After finding the profile, the optional "security classification check" occurs if installed. If this check is installed, a security classification for each subject and object must be defined and is then maintained in the appropriate profile.

This check compares the subject's "security level number" and categories to those of the object's. If either the security level of the subject is less than that of the object, or the subject's category set does not contain all entries in the object's category set, then RACF fails the request. If both tests pass, RACF continues by checking the access authority in the profile[1].

---

[1] Since Security Classification can grant either read or write access to an object dominated by a subject, such an access check can not be used to provide the Mandatory Access Control defined in the TCSEC [40].

Profile Checking

During profile checking, RACF determines if the subject possess the proper access authority by referring to the object's profile. If the user is listed and the access authority is sufficient, RACF permits access.

Otherwise, RACF checks the authority of the group to which the user is presently connected to. If the group is listed in the object's profile and the group has sufficient access authority, RACF permits access. If not, RACF determines if the GRPLIST option is active at the site. The GRPLIST option enables RACF to compare all groups to which the subject belongs. If active and if any of the subject's groups are listed and have sufficient access authority, RACF permits access.

Universal Access Authority

After failing the user and group profile check, the requested access authority is compared to the UACC. The UACC entry defines the type of access to be granted to any member of the system. If the UACC provides sufficient access authority, RACF permits access. For the C2 configuration, the UACC must be set to NONE.

OPERATIONS Attribute Checking

As a last check, RACF will determine if OPERATIONS access is allowed for the object's RACF class. RACF then determines if the subject has the OPERATIONS attribute and permits access if possessed. Otherwise, RACF fails the request.

After RACF completes its access check, a return code is issued to CP. Based upon this return code, CP makes the final access decision. CP abides by RACF's access decision for all objects except minidisks and spool files. For these objects, CP refers to table values established by the SYSGEN macro. This table defines CP's overriding action to take for the FAILED, UNDEFINED, and WARNING return codes from RACF. If anyone of these return codes is present, CP can change the code to FAIL, DEFER, or PASS. This changed result becomes the final decision. For an operational C2 system, FAIL must be selected for the FAILED, UNDEFINED, and WARNING return codes.

Protected Objects

VM/SP with RACF provides DAC for named objects in the following manner:

Minidisks

Minidisks are protected by RACF profiles, either discrete or generic, under the VMMDISK class. Users with the SPECIAL attribute or the CLAUTH attribute to minidisks are able to assign an access authority for each user to each minidisk or minidisks with similar security requirements.

Spool Files

The virtual printer, punch, and reader are the only mechanisms which a subject can use to manipulate the spool files. Access control is therefore performed through these mechanisms. CP will only allow a user to read and write to his own virtual printer and virtual punch, and thus, only the subject's spool file. Virtual readers are protected

by RACF profiles, under the VMRDR class. Thus, other subject's spool files are protected.

Tape Volumes
Tape volumes are protected by resource profiles under the TAPEVOL class. RACF is called by VM-TAPE to determine if the requesting subject is authorized to access the tape volume. RACF protects all defined tape volumes with an IBM standard label or ANSI label.

IUCV Buffers
Specific action on the part of two virtual machines must be initiated to activate this interprocess communication. See page 41, "IUCV" for more information.

VMCF Buffers
For this IPC, the receiving VM must authorize VMCF interrupts and reserve a buffer in virtual memory to store VMCF messages. See page 39, "VMCF" for more information.

VCTCA Channels
The sending and receiving VMs must issue the DEFINE and COUPLE commands respectively to initiate the VCTCA IPC. See page 43, "VCTCA" for more information.

APPC/VM Buffers
Both VMs must authorize the receipt of APPC messages. See page 43, "APPC" for more information.

MSG Buffers
CP software is used to transmit messages initiated by either the MSG or SMSG CP commands. CP uses either VMCF or IUCV IPC to transmit the message, depending the CP Directory entry authorizations. See page 39, "VMCF" and page 41, "IUCV" for more information.

Object Reuse

The storage objects protected by VM/SP with RACF (see page 70, "Objects") are:

Virtual Storage
is the Random Access Memory (RAM) which is accessible to each Virtual Machine. During allocation, CP selects as many 4K pages of unused memory required to meet user's needs. Then these pages are marked used and all memory locations in the page are cleared to binary zeroes. When memory is deallocated, CP marks the memory page as unused. During sysgen all virtual memory pages are initiated as unused. Therefore, memory can only be in three states: used clear from CP, unused from CP, and used written during Virtual Machine use.

Minidisks
: are allocated and deallocated by the Systems Administrator via the dual registration of resources under RACF and DIRMAINT. The System Administrator's portion of the *C2 Security Guide* [28] explains how to set the DMPP option, DISK_CLEANUP = YES, to assure that minidisk extents are cleared before assignment.

Temporary Minidisks
: is the pool of DASD space allocated by the Systems Administrator at sysgen time which is temporarily allocated to a VM for the life of a session. CP clears these resources before they are allocated by calling DIRMAINT to format the temporary disk extents.

Spool Files
: are handled very much like Temporary Minidisks. The Spool File system is allocated DASD by the System Administrator at sysgen time. CP manages this DASD and makes sure that it is cleared each time it is allocated.

Tape Volumes
: are allocated and deallocated by the user. VMTAPE maintains a catalog of all used tape volumes on the system. Each time the user requests a scratch tape to be mounted, the system verifies that the tape is uncataloged. If it is, the tape is cataloged and the user is given access to the tape. When the tape is no longer needed, i.e., the tape's expiration date stamp is less than the current day, the tape volume is uncataloged and the tape is flagged for scratching. The C2 Security Guide notes that the site administration is responsible for degaussing this tape before allowing it to be reused as a scratch tape.

Identification and Authentication

All subjects are authenticated by RACF except the trusted subjects (see page 28, "TCB Components List") that are AUTOLOGed during IPL. Thereafter, all subjects, either trusted or not, are authenticated at log on. This means that if a trusted subject were to logoff or be disconnected, it must authenticate itself to the system.

The authentication process is initiated by entering the LOGON command and the ID of the individual VM. CP identifies the entry and calls RACF. RACF then prompts for a password to authenticate the subject. If an invalid password was entered, then the attempt to access the system fails. If the subject is not defined in the RACF data base, then the subject is unable to access the system.

The following section gives an explanation of RACF groups, user ids', group ids, and password processing.

RACF Groups

A RACF group is a logical combination of users possessing similar access and resource requirements. RACF provides enough flexibility so that an installation can define a group

structure which directly maps to the structure of their environment. This is accomplished by mapping the groups into a tree structure. That is, each group except the highest group, SYS1, has a superior group. Groups can also be a superior group for more than one group. The groups under a superior group are called sub-groups.

All groups are assigned an owner. A group owner can be a RACF defined user or the superior group. Ownership gives the user or group the authority to add or remove users from the group, modify the group profile, define and delete sub-groups, and define new users to RACF if the owner has been assigned the CLAUTH attribute in the USER class. Although the owner has the capability to assign access to the group's resources, the owner is still required to pass the appropriate access checks before access to any resource is granted. In addition, these capabilities can be performed by a user with the SPECIAL attribute, or with the group-SPECIAL attribute but only within the scope of the group.

Group Scope

Resources within the scope of the group are those resources which are owned by the group, owned by users who are owned by the group, and resources owned by all levels of sub-groups that are owned by the group. Therefore, any group authority, either SPECIAL, AUDITOR, or OPERATIONS, assigned to the user encompasses this scope. However, this authority does not extend to resources owned by a group or user that is owned by a user. DSMON is able to produce a group tree report which lists the scope of a group's authority.

As specified above, groups can own another group. Ownership is specified in the group profile and can be changed by assigning another group id or user id in the OWNER operand of the ALTGROUP or ADDGROUP commands for RACF. Since ownership is specified in the profile, it follows that groups can also own user, connect, and other resource profiles.

When an installation defines a group, it can not have the same name as any other group or userid.

Userid, Groupid, and Passwords

Every user attempting to logon to a VM system is required to have a userid and password. A userid is any combination of alphanumeric characters up to eight characters in length. No userid can be the same as any other userid or groupid.

An installation can also restrict a user's ability to logon to the system by specifying certain days of the week and hours within those days that the user can logon. Additionally, a user can be limited to individual terminals on certain days of the week and hours within those days.

During logon, the user can select a group to connect to, by specifying the groupid. A valid groupid is a combination of alphanumeric characters up to eight characters in length and can not begin with a number. Also, No groupid can be the same as any other groupid or userid. If no groupid is specified or if the user does not have a CONNECT profile to the specified groupid, then the user is connected to the user's default group. This information is contained

in the CP directory. A user can only be connected to one group at a time and can only change to another group at logon time. However, if the installation specifies the GRPLIST argument on the SETROPTS command, the user can possess the capabilities of all groups to which the user can connect.

An owner of a group has the option to restrict certain members of the group from logging onto the system from designated terminals. This option, the group terminal option, will override access assigned to the UACC of the terminal's profile.

All users are assigned a password when the userid is defined to RACF. This password acts as a temporary password because it is known by the user and the administrator. For this reason, RACF instructs the user to change the password when the user logs on for the first time.

All passwords must meet rules established by the installation. The PASSWORD operand of the SETROPTS command establishes these rules and can only be executed by a user with the SPECIAL attribute. This user can select the minimum and maximum length of a password and the type of characters that can reside in a position of a password, either a vowel or a number. Also, the interval in which all user passwords remain valid can be selected and after that period the password must be changed. RACF can also store a number of previous passwords in the user's profile so that these passwords can not be used repeatedly. All passwords are stored in the user's profile in either a masked or in an encrypted form.

Password Processing

If the terminal is already powered on, a user can power it off; thus assuring that the user is connected to CP in CP READ state and not to another user's VM. A user begins a logon session by entering the LOGON command and the userid. If the installation is using the password suppression facility, the password must be entered on the next line. Once the userid is entered, CP executes the logon scheduler. This scheduler searches the CP directory for a matching userid. If found, the user is identified and the RACINT macro call is made to authenticate the user. RACF then prompts the user for a password and checks it against the password stored in the user's profile.

During RACINIT processing, the user's specific authority to the terminal and any restrictions within the terminal's RACF profile, if assigned, is also checked. This optional check is called terminal authorization checking. RACF first checks to see if the terminal is restricted from being used at the time of logon. If not, RACF checks the user's profile to see if the user has the authority to access the terminal at that time. The RACF class TERMINAL must be activated for these checks to occur. Also, global access checking is not performed for terminal authorization checking.

If access is denied to the prospective user, then an appropriate audit message is logged and the terminal redisplays the logon screen.

Audit Mechanisms

CP, in conjunction with RACF, can audit the security relevant CP commands, DIAGNOSE instructions, spool activities, and communication between VMs. Additionally, CP can audit any subset of these events for each individual VM. In either case, the events to be audited are specified in an ACI bit map, either individual or system (see page 62, "RACF Interface"). As in an access check, CP references an ACI bit map, specifically the "audit" bit, to determine if RACF needs to be called. If the bit is set for the event, RACF will be called so that an audit record is cut. The complete list of events that can be specified are listed in Appendix A of the *C2 Security Guide* [28]. Users with the RACF AUDITOR attribute can set the auditable event bits in an ACI bit map.

While the individual DIAGNOSE instructions issued by the DIRMAINT and DATAMOVR Service VMs could be audited using these RACF based mechanisms, DIRMAINT itself creates an audit log that is more detailed and informative than the entries that RACF could create. So although the actions of all other SVMs are auditable by RACF, Virtual Machine/Directory Maintenance Licensed Program security relevant events are to be found in the DIRMAINT console log.

RACF always logs the following:

- Every use of the RVARY command, which can deactivate the RACF database, or the SETROPTS command, which allows various RACF options to be set.
- Each time the SETRACF command, which can be used to deactivate RACF, is invoked.
- Each time the RACINIT macro fails to verify a user attempting to access an unauthorized terminal.
- Each time the RACINIT macro fails to verify a user giving an invalid password.

The auditor can direct RACF to log the following types of events:

- Use of the RACDEF macro to define a resource to RACF.
- Any change to a RACF profile as the result of a RACF command.
- Attempts to access RACF protected resources.
- Unauthorized attempts to issue RACF commands reserved for the SPECIAL or group-SPECIAL user.
- RACF commands issued by users with the SPECIAL RACF attribute.
- All RACF-related activities of specified users.
- Access granted by the OPERATIONS check.

Additionally, owners of resources can specify in the resource profile what types of accesses to audit and at what level of access to log (e.g. READ, UPDATE). This type of auditing will appear in the system audit log and can only be superseded by the RACF auditor.

RACF Audit Log

When RACF cuts an audit record, it is written as a Systems Management Facility (SMF) record (an IBM corporate format) to the RACF Audit log. The RACF Audit log, called SMF DATA, is located either on the RACF/VM 301 minidisk or the RACF/VM 302 minidisk which functions as the alternate. RACF determines which minidisk is the active audit log by checking the SMF CONTROL file. Normally, records are written to the 301 minidisk. For RACF audit to execute in accordance with the current TCSEC criteria interpretations, the CLOSE field in the RACF SMF CONTROL file should be set to 001 on the RACF/VM 191 A disk.

RACFSMF is responsible for archiving the minidisk with the SMF DATA file. Upon overflow of the current data file, RACFSMF automatically runs an EXEC which does the following:

- Issues the SMF SWITCH command to activate RACF/VM's alternate SMF DATA minidisk to continue audit recording.
- Initiates an append of the filled audit minidisk to RACFSMF's (archive) 192 minidisk.
- Issues the proper commands to erase the data in the filled audit minidisk. This minidisk is now the alternate SMF DATA minidisk.

Later, the auditor can link to the RACFSMF archive minidisk and execute the RACF report writer to view the audit reports.

The single record for the RACF SMF CONTROL file should include the SEVER = Yes option as indicated in the C2 Security Guide. This option instructs RACFSMF to break the connection between RACF and CP if the current audit minidisk fills before the alternate is available (see page 62, "RACF Interface").

A user assigned AUDITOR or group-AUDITOR attribute can set RACF audit options through the use of the ISPF audit panels or the use of the set RACF options command, SETROPTS.

A user with the AUDITOR attribute has auditor authority over all users and resources defined to RACF. A user with only a group-AUDITOR attribute is restricted to control of RACF auditing for a group and its subgroups. (see page 60, "RACF Groups".)

The summary of SETROPTS operands for the RACF auditor is given below. Each operand also has a negating form, NOoperand, which turns off logging for the following operands:

AUDIT
Log modifications to profiles in the specified class or classes and all uses of the RACDEF SVC.

SAUDIT
Log all commands of SPECIAL and group-SPECIAL users.

OPERAUDIT
Log all successful access to resources and all ADDSD, ALTDSD, RDEFINE, and RALTER commands by OPERATIONS and group-OPERATIONS users who have the OPERATIONS attribute.

CMDVIOL
Log all command violations.

SECLEVELAUDIT
Log access attempts to all RACF protected resources based on the specified security level.

RACF Report Writer

The RACF report writer is used to review pertinent SMF records. The report writer is able to generate a formatted report of attempts to access a particular RACF protected resource, user and group activity, and of system and resource usages.

To use the report writer, the RACF auditor must first link to the minidisk with SMF data on it and then execute the RACRPORT EXEC to invoke it. The VM from which the auditor executes the RACRPORT EXEC must be given link access to the SMF disk. The active minidisk to which SMF data is being recorded by RACF can not be used by the report writer. To use the most current audit data, the auditor must SWITCH the minidisk being used. The report writer executes in the RACMAINT virtual machine or a VM with the same privileges and with access rights to link the appropriate minidisks. Such a VM is commonly set up so that RACMAINT can still be used during audit log processing. RACRPORT executes in three stages: command and subcommand processing, record selection, and report generation. Report Writer selection criteria is read from the RACFRW CONTROL file which is created by the auditor for each run of RACRPORT and stored in a minidisk of the VM on which the RACRPORT EXEC is to run. Afterwards, the report can be saved as a CMS file on a minidisk and will remain until it is overwritten by a subsequent invocation of the report writer, or until it is erased. Procedures exist to save a report file permanently.

Command and subcommand processing begins when the RACRPORT EXEC is invoked by a user with the SPECIAL or AUDITOR attribute. The RACRPORT EXEC invokes the report writer command RACFRW and its available subcommands SELECT, EVENT, LIST, SUMMARY, and END. The commands are used to select which SMF records the report uses during the generation of the requested reports.

During record selection, the RACF report writer compares the selected input to the SMF file. The selected input was specified with the SELECT and EVENT subcommands. The report writer then reformats the data into a human readable form. This data will be further sorted in the report generation step before it is written as a CMS file. The record selection step can be bypassed when the input SMF file has already been sorted.

The report generation step creates the requested reports based upon the LIST and SUMMARY commands. The report writer then saves the report before terminating and returning to the invoker.

Data Security Monitor

The Data Security Monitor, DSMON is intended to report the status of RACF protected resources and can only execute when RACF is active because DSMON calls the RACF manager to obtain information from the data base. DSMON can only be executed by the auditor when the RACDSMON EXEC is invoked. DSMON executes in RACMAINT.

DSMON is able to produce the following types of reports:

System Report
This report displays the configuration of the system including the processor type and software releases installed.

CDT report
The CDT report displays all active RACF classes. It also display whether statistics are being kept of each class, whether auditing is being done, whether a OPERATION user is allowed to access resources in the class, and the default UACC for profiles defined in the class.

GAT report
The GAT report displays all resources listed in the GAT and the required authority level to access the resource.

Group Tree report
This report lists all subgroups for every group.

RACF started procedures
This report is only used on MVS and it lists each entry in the started procedures table.

selected user attribute report
This report lists all RACF user with the SPECIAL attribute.

selected data sets report
This report lists all data sets meeting the selected criterion.

RACF exits report
This report lists the names of all installation defined exits and their size.

Program Properties table report
This report is only used by MVS and it lists all the programs in the program properties table.

Even though some of the reports are only available for RACF running on MVS, they are listed above since they are included in the RACF code supplied for VM/SP with RACF.

DIRMAINT Audit Recording

The DIRMAINT virtual machine maintains the DIRMAINT console log which is a sequential listing of all incoming DIRMAINT commands that altered the source directory. This history file remains active until a prearranged time for backup, at which point it is copied as a backup file and a new one created. The Service VM maintains a default number of backup files, called generations, and once that number is met DIRMAINT begins overwriting the oldest generation. These audit records can be inspected by logging onto the DIRMAINT SVM and using XEDIT, the system edit facility, to inspect them.

Auditing of Service Virtual Machines

Through the use of an individual ACI bit map, it is possible to audit designated actions of any virtual machine. The bit map will allow an installation to record the selected CP commands, VM communications, and/or the Diagnose instructions that those machines may issue. In this way, the commands commonly used by any SVM can be added to the audit logs. Use of RACF commands during a RACF session initiated by a user logged into another VM can be audited by using the selective audit feature, specifying the user's logon VM. This can be used to audit uses of privilege by users who have been assigned RACF attributes or authorities (see page 73, "Privileges").

The OPERATOR VM and any use of VMTAPE-MS can be logged by using selective audit. For a VMTAPE-MS audit event, it is necessary to log the identity of a user who successfully requests VMTAPE-MS to mount a tape. The one security relevant command that VMTAPE-MS executes is the ATTACH command, and RACF logs not only that VMTAPE ATTACHed a tape, but also records for whom the tape was attached. The sequence of RACF commands required to accomplish this is referenced in the *C2 Security Guide* [28].

One may log use of specific commands by the OPERATOR (or any other user, for that matter) by invoking appropriate sequences of RACF commands as discussed in *C2 Security Guide*.

It should also be noted that the DIRMAINT and VMTAPE-MS SVMs have other audit logs which serve specific purposes related to their operations, but which are not related to the selective audit functionality provided by the individual bit map capability of VM/SP with RACF.

ISPF Dual Registration Logging

Since the ISPF menu screens are used to generate both RACF and DMPP commands that are later sent to the corresponding service machines, all actions by system administrators are auditable when those commands reach the service machines. Therefore, commands issued using the ISPF dual registration panels need not be separately audited.

EVALUATION AS A C2 SYSTEM

Discretionary Access Control

Requirement

> The TCB shall define and control access between named users and named objects (e.g., files and programs) in the ADP system. The enforcement mechanism (e.g., self/group/public controls, access control lists) shall allow users to specify and control sharing of those objects by named individuals, or defined groups of individuals, or by both, and shall provide controls to limit propagation of access rights. The discretionary access control mechanism shall, either by explicit user action or by default, provide that objects are protected from unauthorized access. These access controls shall be capable of including or excluding access to the granularity of a single user. Access permission to an object by users not already possessing access permission shall only be assigned by authorized users.

Applicable Features

VM/SP with RACF defines and controls access between virtual machines and named objects through RACF profiles and explicit action of a subject. RACF profiles can specify individual virtual machines or groups to which virtual machines belong, along with an access authority (i.e. None, READ, UPDATE, and ALTER). A subject must specify the name of a virtual machine when taking explicit action.

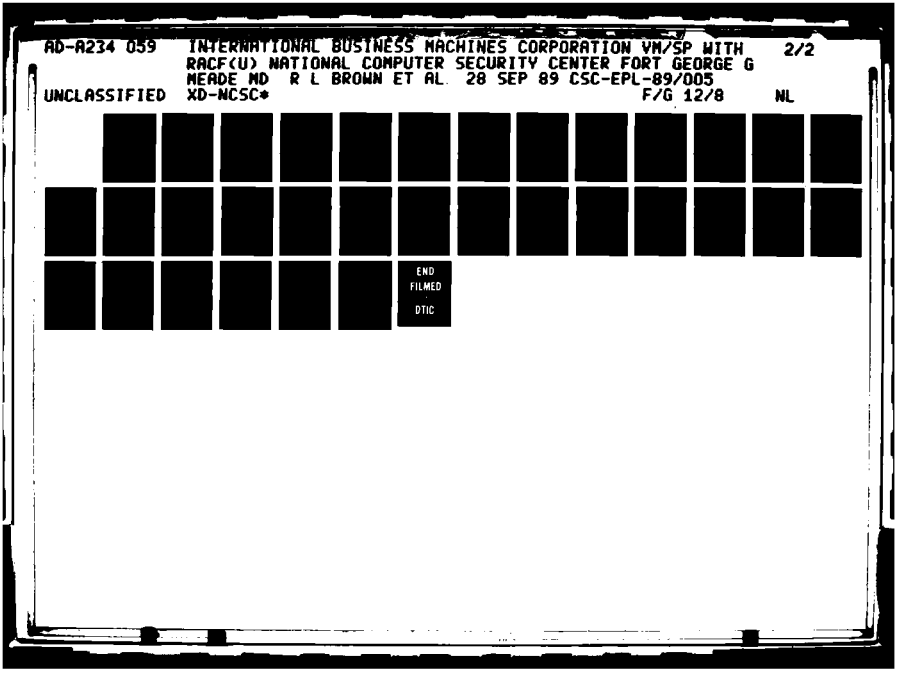
Propagation of the access rights contained within the profiles is restricted to those subjects that possess the ALTER access to an object and those that have profile ownership. Additionally, subjects possessing the SPECIAL or group-SPECIAL attribute can manipulate the profile so that they could possess the ALTER or ownership authority.

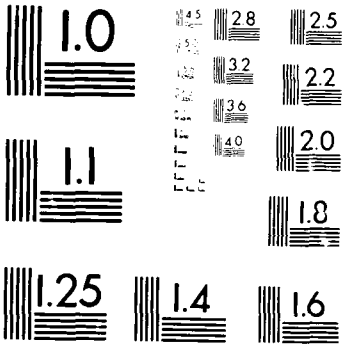
An access authority of NONE is assigned to the UACC. This prohibits unauthorized subjects from gaining access to named objects protected by RACF profiles. Any named object that depends on RACF for discretionary access control that is created without a RACF profile cannot be accessed by any user defined to RACF. Other explicit action on the part of a user or system administrator is required to allow access to other types of named objects. Thus, all named objects are protected from access of any kind by default, unless explicit user action has been taken. The requirement that dual registration be used when creating a new user account ensures that all users are defined to RACF.

A complete description of the DAC mechanisms and how they relate to each named object is provided on page 74, "DAC".

Conclusion

VM/SP with RACF satisfies the C2 Discretionary Access Control requirement.


AD-A234 059 INTERNATIONAL BUSINESS MACHINES CORPORATION VM/SP WITH 2/2
RACF(U) NATIONAL COMPUTER SECURITY CENTER FORT GEORGE G
MEADE MD R L BROWN ET AL. 28 SEP 89 CSC-EPL-89/005
UNCLASSIFIED XD-NCSC* F/G 12/8 NL
END
FILMED
DTIC



Object Reuse

Requirement

> All authorizations to the information contained within a storage object shall be revoked prior to initial assignment, allocation, or reallocation to a subject from the TCB's pool of unused storage objects. No information, including encrypted representations of information, produced by a prior subject's actions is to be available to any subject that obtains access to an object that has been released back to the system.

Applicable Features

For VM/SP with RACF, CP performs the functions which address the object reuse requirement for virtual storage, spool files, and temporary disks. For virtual storage, CP assures that the first reference to each page of the virtual storage, which is what causes the page fault that brings the page into real storage, causes that storage to be overwritten with user data or zeros. For spool files, CP assures that the allocated DASD space is cleared before allocation. For temporary disks, the required setting, SYSRES SYSCLR = YES, in the CP directory assures that CP will overwrite the temporary disk upon allocation. The *C2 Security Guide* explains how minidisks are to be cleared upon initial assignment and that tape volumes must be degaussed upon notification from VMTAPE.

The TCB provides no mechanism to directly manipulate or view data storage within the controller, terminal and other devices which the system controls. Only a hardware oscilloscope can directly read these devices. Terminals do allow users to view the last screen of data; however, all terminals under evaluation have a clear screen key or at least the ability to remove all data from view via a succession of enter keys. Operation of the terminals is described in VM/SP CMS Command Reference [17]. Therefore, the user can prevent others from reading data from a terminal. Printers could be asked to reprint the contents in their buffers from the printer control panel. Since These printers require sufficient procedural security to prevent this. Cluster controllers and disk controllers should also be physically secured.

Buffers and the various caches on the system are transparent to the user virtual machine and are only accessible by components of the TCB. The TCB ensures that user data is kept separate in these buffers and caches.

Conclusion

VM/SP with RACF satisfies the C2 Object Reuse requirement.

Identification and Authentication

Requirement

> The TCB shall require users to identify themselves to it before beginning to perform any other actions that the TCB is expected to mediate. Furthermore, the TCB shall use a protected mechanism (e.g., passwords) to authenticate the user's identity. The TCB shall protect authentication data so that it cannot be accessed by any unauthorized user. The TCB shall be able to enforce individual accountability by providing the capability to uniquely identify each individual ADP system user. The TCB shall also provide the capability of associating this identity with all auditable actions taken by that individual.

Applicable Features

Identification and authentication is controlled by CP and RACF. The CP command LOGON is used to identify subjects to the VM system. When a LOGON command is issued, CP searches for the userid in the CP Directory. If found, RACF is called. RACF uses the userid to find the user's profile and then prompts the user for the password. If the entered password is verified by RACF, the user is authenticated. Only identified and authenticated users can gain access to the system. See page 78, "Ident. and Auth." for details.

Conclusion

VM/SP with RACF satisfies the C2 Identification and Authentication requirement.

Audit

Requirement

> The TCB shall be able to create, maintain, and protect from modification or unauthorized access or destruction an audit trail of accesses to the objects it protects. The audit data shall be protected by the TCB so that read access to it is limited to those who are authorized for audit data. The TCB shall be able to record the following types of events: use of identification and authentication mechanisms, introduction of objects into a user's address space (e.g., file open, program initiation), deletion of objects, actions taken by computer operators and system administrators and/or system security officers, and other security relevant events. For each recorded event, the audit record shall identify: date and time of the event, user, type of event, and success or failure of the event. For identification/authentication events the origin of request (e.g., terminal ID) shall be included in the audit record. For events that introduce an object into a user's address space and for object deletion events the audit record shall include the name of the object. The ADP system administrator shall be able to selectively audit the actions of any one or more users based on individual identity.

Applicable Features

The VM/SP with RACF audit mechanism can log all command execution, generation of output, object access, violations, and object modification events. The audit data is recorded in the RACF SMF DATA file, except that a log of all Virtual Machine/Directory

Maintenance Licensed Program activity is captured in the DIRMAINT console log. These files reside on RACF protected minidisks belonging to Service VMs within the TCB. As such the audit information is protected from unauthorized access, modification and destruction by RACF access rules.

The RACF SMF DATA file can be copied to the archive minidisk by users with AUDIT and SPECIAL privilege. The DIRMAINT console log can be inspected by a user with DIRMAINT Staff privilege. VM/SP with RACF provides users with the AUDIT or SPECIAL privilege access to audit log reduction tools in the form of report generators to allow the review of the RACF SMF audit data. For RACF SMF DATA files, the RACF Report Writer is provided. For especially large output files, the XEDIT editor may be needed to locate particular records. The tool available for inspection of the DIRMAINT console log data is the XEDIT editor.

Conclusion

VM/SP with RACF satisfies the C2 Audit requirement.

System Architecture

Requirement

> The TCB shall maintain a domain for its own execution that protects it from external interference or tampering (e.g., by modification of its code or data structures). Resources controlled by the TCB may be a defined subset of the subjects and objects in the ADP system. The TCB shall isolate the resources to be protected so that they are subject to the access control and auditing requirements.

Applicable Features

Separation of the TCB domain of execution and isolation of the system resources are accomplished by a combination of hardware and software protection features. Dynamic Address Translation, which depends on tables whose entries are set by an element of the TCB, provides isolation of virtual address space. All elements of the TCB are either memory resident portions of CP, or swappable portions of CP which always remain under the control of CP, or else execute in service Virtual Machines which are in distinct address spaces controlled by CP. See page 19, "DAT" for more details.

The hardware provides a two state machine, and only CP runs in real supervisor state. All virtual machines, including those that are part of the TCB, can only simulate supervisor state

and actually run in problem state. Thus, only CP has access to system tables, including those needed for DAT. See page 24, "Hardware Separation and Protection Mechanisms" for more details.

Conclusion

VM/SP with RACF satisfies the C2 System Architecture requirement.

Conclusion

VM/SP with RACF satisfies[1] the B1 System Architecture requirement. Since the service VMs operate in distinct address spaces separated by the use of Dynamic Address Translation under the control of CP, VM/SP with RACF meets this additional requirement.

System Integrity

Requirement

> Hardware and/or software features shall be provided that can be used to periodically validate the correct operation of the on-site hardware and firmware elements of the TCB.

Applicable Features

IBM provides software which validates the correct operation of hardware and firmware elements of the TCB. Each system, including processor, channels, control unit and I/O devices, is checked via microcode diagnostics and one of the following:

- The System Test 370 (ST370) diagnostic program for 370 and the 3031 processors
- The System Test 4300 (ST4300) diagnostic program for 43XX processors
- The New System Test (NST) for the 3032, 3033, and 308X processors
- The Processor Complex Exerciser (PCE) and the Channel Subsystem Exerciser (CSE) for the 3090 series processors
- The System Test 9370 (ST9370) diagnostic program for 937X processors

To protect the integrity of real memory from transient errors in the hardware each page of memory is checked for modification. This is explained in more detail on page XX.

---

[1] Although VM/SP with RACF satisfies this requirement at the B1 level, it does not satisfy the assurance requirements above its rated level.

Conclusion

VM/SP with RACF satisfies the C2 System Integrity requirement.

Security Testing

Requirement

> The security mechanisms of the ADP system shall be tested and found to work as claimed in the system documentation. Testing shall be done to assure that there are no obvious ways for an unauthorized user to bypass or otherwise defeat the security protection mechanisms of the TCB. Testing shall also include a search for obvious flaws that would allow violation of resource isolation, or that would permit unauthorized access to the audit or authentication data.

Applicable Features

IBM has submitted a test plan document [10] containing test procedures for vendor testing performed on the security features of VM/SP with RACF. These tests have been run on one of the hardware configurations identified in the IBM test plan. The results have been supplied to the evaluation team.

This material was reviewed by the evaluation team as part of the evaluation of the C2 Test Documentation Requirement. For discussion of this review please see page 95, "Test Documentation".

The evaluation team developed an Evaluation Team Test Plan [41] consistent with the Guideline for Security Testing given in the *Trusted Computer System Evaluation Criteria* [40].

System Testing

A successful test readiness review was held three weeks prior to system security testing. It was attended by the team test director and another team member. Hardware integrity tests were described. A major subset of the vendor tests were run. These tests were run on a 4361 configuration comparable to the actual test configurations.

Testing was performed by the full team on a 4381 (UP/NON-HPO) configuration and on a 3090 (MP/HPO) configuration. The complete hardware and software in each configuration is given in Appendix C.

Upon arrival the team inspected the test environment and documented the two test configurations. Hardware integrity tests were run for a total of more than four hours for each configuration. These integrity tests covered CPU instructions, main storage, channel paths and devices. No hardware errors were detected during these tests.

From the information in the *C2 Security Guide* [28] and referenced systems manuals, a C2 VM/SP with RACF for a uniprocessor (UP) was generated for the 4381. Then each of the additional components of the TCB were generated from corresponding distribution files. This system was copied to the 3090 and a two-processor (MP) system was generated which included HPO and which was also consistent with the settings of the *C2 Security Guide.*

Once each system had been completely installed, the team observed execution of the functional test suite on each test configuration. The CMS regression tests were not run on

the 3090 configuration. The CMS tests ran correctly on the 4381 configuration. All other tests in the vendor test suite were run on both configurations.

Additional Tests

Although the team inspected the vendor test plan and is confident that the tests cover all security relevant events, the team expected to run several additional tests. Some of these additional tests were not run, since it was determined that the objective of the test was completely satisfied by one or more of the vendor tests. The detail of each team test is given in Appendix C.

All vendor tests and all additional team tests that were attempted ran to completion, and ran as expected. Where applicable, tests were run on both test configurations.

Conclusion

VM/SP with RACF satisfies the C2 Security Testing requirement.

Security Features User's Guide

Requirement

> A single summary, chapter, or manual in user documentation shall describe the protection mechanisms provided by the TCB, guidelines on their use, and how they interact with one another.

Applicable Features

The *Virtual Machine/System Product C2 Security Guide* [28] is a three part document which provides a mapping to the documents that comprise both the *Trusted Facility Library and Security Features User's Library for VM/SP with RACF.* Part One provides information for every user and administrator on the system, while Part Three discusses the special restrictions, responsibilities, and requirements placed upon the general user.

The following manuals comprise the Security Features User's Library and collectively provide sufficient restrictions, responsibilities, and requirements for the general user of a VM/SP C2 TCB:

- "Part Three: For General Users" in *C2 Security Guide* [28]
- the VM HELP Facility [29]
- RACF General User's Guide [32]
- VM/SP CMS Command Reference [21]
- DIRMAINT Operation and Use [23]
- VMTAPE-MS User's Guide [39]

Conclusion

VM/SP with RACF satisfies the C2 Security Features User's Guide requirement.

Trusted Facility Manual

Requirement

> A manual addressed to the ADP system administrator shall present cautions about functions and privileges that should be controlled when running a secure facility. The procedures for examining and maintaining the audit files as well as the detailed audit record structure for each type of audit event shall be given.

Applicable Features

The *Virtual Machine/System Product C2 Security Guide* [28] is a three part document which provides a mapping to the documents that comprise both the *Trusted Facility Library and Security Features User's Library for VM/SP with RACF*. Part One provides information for every user and administrator on the system, while Part Two also explains Security and Integrity Enhancements which allows for more opportunity to audit CP commands, DIAGNOSE functions, and virtual machine communication.

The following manuals, which are referenced in the C2 Security Guide referred to above, comprise the Trusted Facility Library and collectively provide a system administrator a good description of what a C2 system is and how the system may be installed and maintained in a secure fashion:

- "Part Two: For Administrators" beginning on page 17 of the guidebook [28]
- the VM HELP Facility [29]

- VM/SP Installation Guide [24]
- VM/SP HPO Installation Guide [36]
- VM/SP CMS Command Reference [31]
- VM/SP System Facilities for Programming [27] VM/SP CP for SystemProgramming [25]
- System Programming Library: RACF [34]
- RACF Program Directory for VM Installations [12]
- RACF Command Language Reference [30]
- RACF Security Administrator's Guide [31]
- RACF Auditor's Guide [33]
- RACF Audit and Control for Individual VM Users [13]
- RACF Diagnosis Guide [14]
- VMTAPE-MS Installation Guide [37]
- VMTAPE-MS Administrator's Guide [38]
- DIRMAINT Operation and Use [23]
- ISPF and ISPF/PDF Installation and Customization [35]

Conclusion

VM/SP with RACF satisfies the C2 Trusted Facility Manual requirement.

Test Documentation

Requirement

> The system developer shall provide to the evaluators a document that describes the test plan, test procedures that show how the security mechanisms were tested, and results of the security mechanisms' functional testing.

Applicable Features

IBM has developed test plans and procedures for testing the security mechanisms of VM/SP with RACF. A test plan document [10] and test procedures have been received and reviewed by the evaluation team. IBM has run these test procedures and supplied the evaluation team with listings of the results. The identified tests, the procedures defined for each test and the results are summarized in the paragraphs that follow.

Vendor Test Plan

The test plan submitted covers security testing for the complete VM/SP with RACF system. IBM identified four hardware configurations drawn from the EVALUATED HARDWARE COMPONENTS listed in Appendix A.

The software components identified in the test plan are CP, RACF, DIRMAINT, VMTAPE-MS, CMS, ISPF and the dual registration panels.

Vendor Philosophy of Test

The security relevant TCB components were subjected first to the four tests described below:

- Unit Test
- Component Test
- System Test
- System Regression Test

Then each component was reviewed against the C2 Class requirements of the TCSEC. [40]

The following procedures were followed in developing informal tests leading to a System Regression Test. Note: Once a unit test is complete, the code for the unit is turned over to an independent Product Control Group (PCG), which manages the code for the rest of the internal test phases.

Unit Test

Each unit of code in a component is subjected to a code review followed by "white box" tests by its programmer. The intent of the review and tests is to test the internal flow of the code.

Component Test

A group, independent of the development group, identifies functional test areas and develops a detailed component test plan, delineating each test case that it will run. As code units are received from the PCG, the developed test cases are run on the code. These tests are to serve as "grey/black box" tests of the functional intent of the code

units as defined by the Final Programming Functional Specification for this component. [8] [3] [7]

System Test
Code for this test is installed in common operational system environments, typical of the most common types of customer installations. Various program products are installed to exercise the components and interfaces to these components from the program products.

System Regression Tests
The final internal test consists of running previous functional tests derived for earlier releases of the system to verify that the security changes implemented in the components have not regressed the existing functionality of previous releases.

Vendor Security Relevant Tests

The second major phase in system testing addresses the security relevant features of each component. Each software component is reviewed and security relevant events are identified. Tests are constructed to verify that each security feature works. The set of tests derived for a component to verify correct operation of a security feature is called a "Testbucket". A two phase process was used to derive the test cases: Variation Identification (to identify features to test) and Test Case Development (to develop a test scenario for each security feature identified for testing). A summary of the testbuckets identified for each of the TCB components is given below:

CP testbucket summary
Twenty-three tests were developed to test twenty-one security relevant events identified for the CP system component. The twenty-one security relevant events are command auditing, diagnose code auditing, spoolfile auditing, SPTape auditing, APPC auditing, IUCV auditing, VMCF auditing, TDisk reuse, CP command classes, CP diagnose classes, CP logon passwords, CP link passwords, storage key protection, shared storage protection, privilege class overrides, password suppression, programmable operator, DAC on IUCV, DAC on APPC, DAC on VMCF, and use of VCTCA protocol.

CMS testbucket summary
CMS is included in the TCB since it is the support service machine for both RACF and VMTAPE-MS. For these two TCB components, CMS provides the interface with CP and other TCB components. The security relevant aspect of CMS is that its basic support functions must perform correctly. IBM has collected a set of system regression tests used to demonstrate that new system changes have not affected the functionality of CMS. These tests will be rerun to show that security features of RACF and VMTAPE-MS are not impacted by CMS.

RACF testbucket summary
Six tests were developed to test six (security relevant events identified for the RACF system component. The six security relevant events are unauthorized use of AUDIT, unauthorized use of 'controllable' events, unauthorized Use of SETEVENT, use of CP

LINK command, restrictions imposed by ACIGROUP statement, and selective audit of CP commands issued by specific users.

DIRMAINT testbucket summary
Three tests were developed to test five security relevant events identified for the DIRMAINT system component. The five security relevant events are staff commands, substaff commands, owner commands, monitor commands and operator commands.

VMTAPE-MS testbucket summary
Three tests were developed to test three security relevant events identified for the VMTAPE-MS system component. The three security relevant events are CATALOG command, MOUNT command, and LIST command.

ISPF and Dual Registration Panel testbucket summary
ISPF controls the display of the logon menu and the capture of the user identification and authentication data. The dual registration panel ensures that redundant user registration information is recorded in the RACF data base and the CP Directory. IBM has submitted one test to test two security relevant events associated with this administrative interface. The two security relevant events are proper registration of a user (VM) and concurrent update of the CP Directory DIRMAINT and RACF data base.

The IBM test plan describes the procedures to follow for each of the tests developed.

Vendor Test Results

Listings have been supplied to the evaluation team of runs made for each identified test case.

Conclusion

VM/SP with RACF satisfies the C2 Test Documentation requirement.

Design Documentation

Requirement

Documentation shall be available that provides a description of the manufacturer's philosophy of protection and an explanation of how this philosophy is translated into the TCB. If the TCB is composed of distinct modules, the interfaces between these modules shall be described.

Applicable Features

The philosophy of protection of VM/SP with RACF is described in *VM C2 Philosophy of Protection* [1]. The interface between the TCB hardware and the software components of the TCB is explained in *System 370 Principles of Operation*[11].

The design documentation that describes how CP works is contained in *System Logic and Problem Determination Guide Volume 1 (CP)* [16] for the standard version of VM/SP, and in *VM/SP High Performance Option System Logic and Problem Determination Guide-CP* [18] for the High Performance Option version of CP. The modifications made to standard CP, in a so called Small Programming Enhancement (SPE), are described in *VM/SP CP C2 Security Enhancements SPE, Final Programming Functional Specifications*[8].

Design documentation for the other components of the TCB are:

- CMS Problem Logic Determination Guide [17]
- CMS Data Areas [19]
- Final Programming Functional Specifications, Resource Access Control Facility, Release 1.8.2 [7]
- Final Programming Functional Specification, Development APAR for C2 Selective Audit/Control, VM/SP Release 5 and Release 6 [6] which describes the new audit facility addition.
- RACF/VM Support [4] which describes extensions to CMS that allow the version of RACF running in the RACF/VM to interact with CP.
- CST Version 4.0 [5] which provides additional information on the CMS extensions in the above reference.
- DIRMAINT Version 1 Release 4 Final Programming Functional Specifications [3]
- Virtual Machine/Directory Maintenance Licensed Program Diagnosis Reference, Release 4 [15]
- VMTAPE-MS, Version 4.1 Program Logic Manual [2]
- ISPF Program Logic Manual [9]

Conclusion

VM/SP with RACF satisfies the C2 Design Documentation requirement.

This page intentionally left blank.

EVALUATOR COMMENTS

Audit Reduction Tools

Although the software architecture of VM/SP with RACF allows a multiplicity of audit logs to be kept by Service VMs, good system security administration is dependent on being able to list all security relevant events in time order of their occurrence. VM/SP with RACF achieves this for the most part, since proper use of the VMEVENT profile in conjunction with selective audit of individual Service VMs allows the RACF Report Writer to create such a list. Unfortunately, the RACF Report Writer audit reduction tool cannot create reports with as fine a granularity as the VMEVENT profile can specify audit records. Careful use of the Report Writer will produce reports small enough to be easily inspected by hand, or by using the XEDIT facility. The extensive discussion of the RACF Report Writer in the *RACF Auditor's Guide* [33] should be studied carefully to take fullest advantage of the Report Writer's capabilities. Setting up a VM with read access to RACFVM's audit log minidisks and execute access to the Report Writer software is recommended. Then a file called RACFRW of type COMMAND must be created for each report, since this is the unchangeable default input file for the Report Writer. COMMAND files for commonly run reports could be created and renamed to RACFRW COMMAND when the auditor needs to run them, then changed back to their original names after the report is generated.

The single exception of Virtual Machine/Directory Maintenance Licensed Program, which has its own log which must be inspected with a tool other than RACF Report Writer, does not violate the audit requirement. The events recorded in the DIRMAINT history log are mainly additions or deletions of subject Virtual Machine specifications and object descriptions, and their exact time of occurrence is not usually that important to an audit trail. If the system administrator suddenly notices a new USERID accessing minidisks, a quick check of the DIRMAINT history log will show when the USERID was created and what privileges that user has. Actually, the administrator is more likely to issue a DIRM QUERY about the user, or run DSMON to find out about the USERID; the DIRMAINT history log will seldom be used except to track the creation of inappropriate USERIDs or minidisks, in which case there is still no need to integrate its contents with the output of the RACF Report Writer.

Batch Capabilities

There are at least two software packages available to provide batch facility to allow users of VM/SP with RACF to execute and monitor time consuming jobs while using their login VM for interactive computing. These packages include VMBATCH and CMSBATCH. Both work by starting a Service Virtual Machine which receives a job stream describing work to be done from a user's VM, and executes the job on the user's behalf, using that user's access permissions.

The vendor has decided not to include either of these systems in the evaluated product, since inclusion of either would require that the SVM be part of the TCB. It has previously been shown that a user can alter the job stream sent to the CMSBATCH SVM after any access checks were performed by it, so inclusion of CMSBATCH would require a major rewrite of the software. While there are no known problems with VMBATCH, questions of

discretionary access and auditing would require extensive research before the team would have the assurance that VMBATCH could be trusted to run in another user's behalf.

The absence of these formal batch processing techniques is not a serious problem, since VM/SP with RACF was designed and is normally used as an interactive system. The availability of EXECs that can be invoked from CMS, and the ability to detach a VM and leave it running, provide a similar capability to the user.

VTAM and non-ASCII Terminals

VM/SP supports the use of VTAM and other forms of terminal communication. The vendor chose not to include these components in the TCB of the evaluated system. Therefore only terminals supported by CP (via Bisynchronous communications (BSC)) directly can be added to the system. These terminals are listed in the appendix. This means that SNA is not supported.

MVS as a Guest Operating System

It has been noted several times in this report that any program that will run on a standalone System/370 can be run in a VM. In addition to CMS, one can run transaction processing systems, database managers, bare machine application programs, or other System/370 operating systems. In one of the most common instances, one can run another VM[1]. One can also run the MVS operating system.

There are many systems that run MVS as a guest operating system, since that allows software maintenance to be performed on an alternate configuration of MVS running in another VM. Once the new configuration of MVS is ready, a switch can be made that will minimize the downtime apparent to the system users. Unfortunately, MVS is a large system and runs slowly in a regular VM due to its performing its own process scheduling and memory management on top of CP's process scheduling and memory management. As a result, systems that normally run MVS lock it into low storage using the VIRTUAL=REAL option, thus preventing CP from controlling either storage references or Channel Command Programs. In addition, since MVS is a multi-user operating system, individual users must use the DIAL command to get the attention of MVS, which then invokes its own login facility. Thus, VM/SP with RACF is unable to perform Identification and Authentication on the users who are known to MVS.

The evaluated configuration of VM/SP with RACF does not allow the VIRTUAL=REAL option, the use of the DIAL command, or the writing of uninterpreted Channel Command Programs. Sites that want to run a secure version of MVS should check the Evaluated Products List and purchase such a system on that list. Sites that require the flexibility of VM/SP with RACF should consider purchasing the evaluated system described here.

---

[1] This is called a second level VM.

EVALUATED HARDWARE COMPONENTS

The hardware list below contains all hardware components that may appear in evaluated configurations of VM/SP with RACF (NO HPO upgrade).

Processors:

Type	Model	Notes	Type	Model
370	135-3		3042	AP
370	138			
370	145-3		3081	D16
370	155-II			
370	158	UP/AP/MP	4321	
370	158-3		4331	2,3,5
370	165-II		4341	
370	168	UP/AP/MP	4361	2,3,5
370	168-3		4381	1,2,3,11,12, 13,14
3031		UP/AP		
3032		UP	9370	30,50,60,70,80
3033		UP/AP/MP	9373	30S
3033	N,S	UP/AP/MP	9375	50,60,70
			9377	80,90S

Dasd: Controllers			Devices	
Type	Model	Notes	Type	Model
2835	1,2		2305	1,2
2844			2314	
3830	1,2,3		2319	
3880	1,2,3		3310	
3880	11,13		3330	1,2,11
3990			3333	1,11
IFA	4650	370/135,370/145	3340	A2,B1,B2
IFA	4655	370/135,370/135-3, 370/138	3350	A2,A2F,B2
			3370	A1,A2,B1,B2
IFA	4660	370/145,370/145-3, 370/148	3375	
			3380	AD4/BD4 AE4/BE4
ISC	4660	370/158,370/168	9332	
			9335	

The hardware list of components that may appear in evaluated configurations of VM/SP with RACF (NO HPO upgrade). (CONTINUED)

Tape: Controllers			Devices	
Type	Model	Notes	Type	Model
2803			2401	
2804			2402	
3411		Controller and device	2403	
3422		Controller and device	2404	
3430		Controller and device	2415	1,2,3,4,5,6
3480		Controller and device	2420	5,7
3803			3410	1,2,3
			3411	1,2,3
			3420	3,4,5,6,7,8
			8809	
			9347	

Printers:

Type	Model	Notes
1403	2,3,7,N1	
1443	N1	With 144 print positions
3203	4,5	Only on a 370/138, 370/148
3211		Right indexing only
3213		in 3215 emulator mode
3262	1,5,11	
3268	2,2C	
3287	1,1C,2,2C	
3289	4	
3800	1,3,8	
4245		
4248	1	
4250		Dedicated Only
5210		
6262		

The hardware list of components that may appear in evaluated configurations of VM/SP with RACF (NO HPO upgrade). (CONCLUDED)

Terminals:

Controllers		Devices			Devices		
Type	Model	Type	Model	Notes	Type	Model	Notes
3174		1050			3215	1	Console
3272	2	2150		Console	3232	51	
3276	2,3,4	2741			3250	1,2	
3274	1B,1D,21A,	3036	1,2		3275	2	
	21B,21D,	3066			3276	2,3,4	
	31A,31D,	3101			3277	2,3,4	
	41A,41D	3178			3278	2,3,4,5	
	51C,61C	3179			3278	2A	Console
3708		3180			3279	2A,2B,3A,3B	
3705		3192			3279	2C	Console
3708		3210	1,2	Console	3290	XC,XD	
3720					3767		
3725					7412	1	Console

The hardware list below contains all hardware components that may appear in evaluated configurations of VM/SP with RACF with the HPO upgrade.

Processors:

Type	Model	Notes	Type	Model	Notes
370	155-II		3090	120E, S	
370	158	UP/AP/MP	3090	150	
370	158-3		3090	150E, S	
370	165-II		3090	170S	
370	168	UP/AP/MP	3090	180	
370	168-3		3090	180E, S	
			3090	200	
3031		UP/AP	3090	200E, S	
3032		UP	3090	250S	
3033		UP/AP/MP	3090	280E, S	
			3090	400	
3042	2	AP	3090	400E, S	
3081		Dyadic	4341		
3083		UP	4381	1,2,11,12,13	UP
3084		Partitioned Processing	4381	3,14	Dual

Dasd: Controllers		Devices		Devices	
Type	Model	Type	Model	Type	Model
2835	1,2	2305	1,2	3350	A2,A2F,B2
3830	1,2,3	3310		3370	A1,A2,B1,B2
3880	1,2,3,11, 13	3330	1,2,11	3375	
		3333	1,11	3380	AA4 AD4/BD4 AE4/BE4
3990		3340	A2,B1,B2	3880	11,13,21,32

Tape: Controllers		Devices	
Type	Notes	Type	Model
2803		2401	
2804		2402	
3411	Controller and device	2403	
3422	Controller and device	2404	
3430	Controller and device	3410	1,2,3
3480	Controller and device	3411	1,2,3
3803		3420	3,4,5,6,7,8

The hardware list of components that may appear in evaluated configurations of VM/SP with RACF with the HPO upgrade (CONCLUDED).

Printers:

Type	Model	Notes
1403	2,3,7,N1	
1443	N1	With 144 print positions
3203	4,5	Only on a 370/138, 370/148
3211		Right indexing only
3213		in 3215 emulator mode
3262	1,5,11	
3287	1,1C,2,2C	
3289	4	
3800	1,3,8	
4245		
4248	1	
4250		Dedicated Only
5210		

Terminals:

Controllers		Devices			Devices		
Type	Model	Type	Model	Notes	Type	Model	Notes
3174		1050			3215	1	Console
3272	2	2150		Console	3232	51	
3276	2,3,4	2741			3250	1,2	
3274	1B,1D,21A,	3036	1,2		3275	2	
	21B,21D,	3066			3276	2,3,4	
	31A,31D,	3101			3277	2,3,4	
	41A,41D	3178			3278	2,3,4,5	
	51C,61C	3179			3278	2A	Console
3708		3180			3279	2A,2B,3A,3B	
3705		3192			3279	2C	Console
3708		3210	1,2	Console	3290	XC,XD	
3720					3767		
3725					7412	1	Console

This page intentionally left blank.

EVALUATED SOFTWARE COMPONENTS

TCB Software Components

- VM/SP Release 5 with C2 Security new function apar VM33580 , or
- VM/SP HPO Release 5 with C2 Security new function apar VM33580
- Conversational Monitor System (CMS), Release 5
- RACF version 1.8.2
- Directory Maintenance Program Product (DMPP) Version 1.4 with support apar VM33536 (VM/SP) or support apar VM33536 and apar VM33114 (VM/SP HPO)
- VMTAPE-MS Release 4.1
- Interactive System Productivity Facility, ISPF Release 2.2

C2 Security Service Level

To insure that the purchaser of VM/SP with RACF has received the same system that was tested by the National Computer Security Center evaluation team, one may check the identification numbers on the feature tapes that are shipped to the site. The appropriate feature tape identification numbers for each of the products listed above appear here:

	Feature Codes				Description
Product	1600	6250	3480	9346	
VM/SP	5359	5365	5366	5367	C2 feature tape
	5012	5013	5025		FBA Starter system + C2 feature
	5026	5027	5028		3330 Starter system + C2 feature
	5039	5040	5062		3350 Starter system + C2 feature
	5063	5071	5072		3375 Starter system + C2 feature
	5081	5042	5054		3380 Starter system + C2 feature
	5355	5356	5357		9313 Starter system + C2 feature
				5358	0671 Starter system + C2 feature
HPO 5		5012	5013		C2 feature tape
		5017	5018		3350 Starter system + C2 feature
		5020	5071		3375 Starter system + C2 feature
		5026	5027		3380 Starter system + C2 feature
RACF	5510	5511	5512	5514	C2 feature tape
DIRMAINT	5810	5811	5812		C2 feature tape

VMTAPE-MS	5870	5871	5872	C2 feature tape
ISPF	5264	5247	5872	C2 feature tape

TCB SVMs

The following Service VMs must be resident on the evaluated system, after being built from the evaluated software.

- AUTOLOG1 Service Virtual Machine
- AUTOLOG2 Service Virtual Machine
- MAINT Service Virtual Machine
- OPERATNS Service Virtual Machine
- RACF Service Machine
- RACMAINT Service Machine
- RACFSMF Service Machine
- DIRMAINT Service Machine
- DATAMOVR Service Machine
- VMTAPE Service Machine
- ISPVM Service Machine

Non-TCB Software Components

- Group Control System (GCS), shipped with Release 5, contains a named, shared segment in storage that one can IPL and run in a user's VM. GCS allows a limited multi-tasking function within a VM which can be used instead of CMS. Since none of the system trusted subjects use GCS, it is not security relevant.
- Transparent Service Access Facility (TSAF), shipped with Release 5, provides a VM to handle APPC/VM connects to resources within a TSAF collection of machines. Since a TSAF collection is a network and not part of the evaluated configuration, only the local features of TSAF are available if this software is loaded into a SVM.
- Environmental Record Editing and Printing (EREP), shipped with Release 5, provides a log of hardware errors when its SVM, EREP, is IPLd.

Interactive Problem Control System (IPCS), shipped with Release 5, provides an interactive, online facility for reporting and diagnosing software failures and for managing problem information status. It uses the output of DUMP information created by CP ABEND or VMDUMP as its input. Since it only looks at the contents of the virtual storage of a single VM, it is not security relevant.

This page intentionally left blank.

REFERENCES

[1] VM C2 Philosophy of Protection, A. J. Nadalin, J. A. Thompson, 5 September, 1989.

[2] VM-TAPE Program Logic Manual, Systems Center Inc., 2 May, 1988.

[3] DIRMAINT Version 1 Release 4 Final Programming Functional Specifications, J. Coyle, W. R. Deniston, Mark J. Lorenc, B. Pederson, 24 August, 1988.

[4] RACF/VM Support, Alex Scianna, 16 September, 1988.

[5] CST Version 4.0, Alex Scianna, 16 September, 1988.

[6] Final Programming Functional Spedification, Development APAR for C2 Selective Audit/Control, Release 5 and Release 6, 14 April, 1989.

[7] Final Programming Function Specifications, Resource Access Control Facility, Release 1.8.2, November, 1987.

[8] VM/SP CP C2 Security Enhancements SPE Final Programming Functional Specifications, S. L. Davis J. Capwell, R. Bargar, March, 1988

[9] ISPF Program Logic Manual, undated.

[10] VM/SP - RACF TEST PLAN FOR C2 DOD EVALUATION, August, 1989

[11] System 370 Principles of Operation, GA22--7000-10, September, 1987.

[12] RACF Program Directory for VM Installations, GC28-1034

[13] RACF Audit and Control for Individual VM Users, GC28-1036

[14] RACF Diagnosis Guide, LC28-1344

[15] Virtual Machine/Directory Maintenance Licensed Program Diagnosis Reference, Release 4, LY20-0889-5, March, 1989.

[16] Virtual Machine/System Product System Logic and Problem Determination Guide Volume 1 (CP), LY20-0892-4, December, 1986.

[17] Problem Determination Vol. 2 (CMS) LY20-0893

[18] Virtual Machine/System Product High Performance Option System Logic and Problem Determination Guide-CP", LY20-0897-8, January, 1986.

[19] Data Areas and Control BlocksVol. 2 (CMS) LY24-5221

[20] Resource Access Control Facility (RACF) Program Logic Manual, Version 1, Release 8, LY28-0730-6, December, 1987.

[21] VM/SP CMS Command Reference, SC19-6209.

[22] VM/SP CP Command Reference, SC19-6211-4, December, 1986.

[23] VM/Directory Maintenance Operation and Use, SC23-0437, March, 1989

[24] VM/SP Installation Guide, SC24-5237, December, 1986

[25] VM/SP CP for System Programming, SC24-5285.

[26] VM/SP Transparent Services Access Facility Reference, SC24-5287-0, December, 1986.

[27] VM/SP System Facilities for Programming, SC24-5288, December, 1986

[28] Virtual Machine/System Product C2 Security Guide VM/SP Release 5 and VM/SP HPO Release 5. SC24-5384-01

[29] The VM HELP Facility (online function)

[30] RACF Command Language Reference, SC28-0733, December, 1987

[31] RACF Security Administrator's Guide, SC28-1340, December, 1987

[32] RACF General User's Guide, SC28-1341, December, 1987

[33] RACF Auditor's Guide, SC28-1342, December, 1987

[34] System Programming Library: RACF, SC28-1343, December, 1987

[35] ISPF and ISPF/PDF Installation and Customization, SC34-4015

[36] VM/SP HPO Installation Guide, SC38-0107, August, 1987

[37] VMTAPE-MS Installation Guide, SH20-6241, September, 1988

[38] VMTAPE-MS Administrator's Guide, SH20-6242, September, 1988

[39] VMTAPE-MS User's Guide, SH20-6245, September, 1988

[40] Department of Defense Trusted Computer System Evaluation Criteria, DOD 5200.28-STD, December, 1985.

[41] IBM VM/SP with RACF Evaluation Team Test Plan, July, 1989.

[42] R. E. Filman, D. P. Friedman, Coordinated Computing - Tools and Techniques for Distributed Software. McGraw-Hill, New York, 1984.

This page intentionally left blank.

GLOSSARY

ACI	Access Control Interface
ADP	Automatic Data Processing
ADSP	Automatic Data Set Protection
ANSI	American National Standard Institute
APPC	Advanced Program to Program Communication
ASCII	American Standard Code for Information Interchange
BC	Basic Control
BLP	Bypass Label Processing
CAW	Channel Address Word
CST	CMS Sub Tasking
CSW	Channel Status Word
CCW	Channel Control Word
CDT	Class Descriptor Table
CDS	Compare Double and Swap
CMS	Conversational Monitor System
CP	Central Processor
CPU	Central Processing Unit
CS	Compare and Swap
CSE	Channel Subsystem Exerciser
CTSS	Compatible Time-Sharing System
DAC	Discretionary Access Control
DASD	Direct Access Storage Device
DAT	Dynamic Address Translation
DCSS	Discontiguous Saved Segment
DMPP	Directory Maintenance Licensed Program
DoD	Department of Defense
DPA	Dynamic Paging Area
DSMON	Data Security MONitor
EC	Extended Control
EPL	Evaluated Products List
FSA	Free Storage Area
GAT	Global Access Table
G	Gigabyte
HPO	High Performance Option
IBM	International Business Machines
IPC	Interprocess Communication
IPL	Initial Program Load
ISPF	Interactive System Productivity Facility
IUCV	Inter User Communications Vehicle
K	Kilobyte
M	Megabyte
MP	Multi Processor
MVS	Multiple Virtual Storage
NCSC	National Computer Security Center
NL	No Label

NST	New System Test
OS	Operating System
PCE	Processor Control Exerciser
PCG	Product Control Group
PSW	Processor Status Word
PUT	Program Update Tape
RACF	Resource Access Control Facility
RAM	Random Access Memory
RAS	Reliability Availability Serviceability
SIO	Start Input Output
SIOF	Start Input Output Fast
SMF	System Management Facilities
SPE	Small Programming Enhancement
SVC	SuperVisor Call
SVM	Service Virtual Machine
TCB	Trusted Computing Base
TCSEC	Trusted Computer System Evaluation Criteria
TFM	Trusted Facility Manual
TLB	Translation Lookaside Buffer
TMC	Tape Management Catalog
UACC	Universal Access Authority
UP	Uni Processor
VCTCA	Virtual Channel to Channel Adapter
VM	Virtual Machine
VMCF	Virtual Machine Communication Facility
VM/SP	Virtual Machine/System Product
VS	Virtual Storage
VTAM	Virtual Telecommunications Access Method
XA	Extended Architecture

☆U.S.GOVERNMENT PRINTING OFFICE:1990-720-120/10961

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; Distribution Unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CSC-EPL-89/005	5. MONITORING ORGANIZATION REPORT NUMBER(S) ~~CSC/120~~ CSC/120
6a. NAME OF PERFORMING ORGANIZATION National Computer Security Center 6b. OFFICE SYMBOL (If applicable) C12	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code) 9800 Savage Road Ft. George G. Meade, MD 20755-6000	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.
11. TITLE (Include Security Classification) Final Evaluation Report IBM VM/SP	
12. PERSONAL AUTHOR(S) Brown, R. Leonard [Aerospace Corp]; Vane, Kenneth D., Gill, David L. [MITRE Corporation]; Oehler, Michael J., Willinghan, Robin A.	
13a. TYPE OF REPORT Final 13b. TIME COVERED FROM ____ TO ____	14. DATE OF REPORT (Yr, Mo., Day) 890928 15. PAGE COUNT 125
16. SUPPLEMENTARY NOTATION	

17 COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	NCSC DAC

19. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The security protection provided by the International Business Machines Corporation VM/SP with RACF operating system software as described in Appendix B, running on one of the IBM 370 processors listed in Appendix A and configured in an appropriately trusted manner as described in the Trusted Facility Manual has been examined by the National Computer Security Center (NCSC). The security features of VM/SP with RACF were examined against the requirements specified by the DOD Trusted Computer System Evaluation Criteria (the Criteria or TCSEC) dated 26 December 1985 in order to establish a candidate rating. The NCSC evaluation team has determined that the highest class at which VM/SP with RACF satisfies all the specified requirements of the Criteria is class C2. A system that has been rated as being a C2 class system provides for discretionary access control (DAC), auditing of security relevant events, and resource isolation.

This report documents the findings of the formal evaluation of the IBM VM/SP with RACF operating system.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL DENNIS E. SIRBAUGH	22b. TELEPHONE NUMBER (Include Area Code) (301)859-4458 8b. OFFICE SYMBOL C12